Département Informatique de l’IUT de l’Université Bordeaux 1

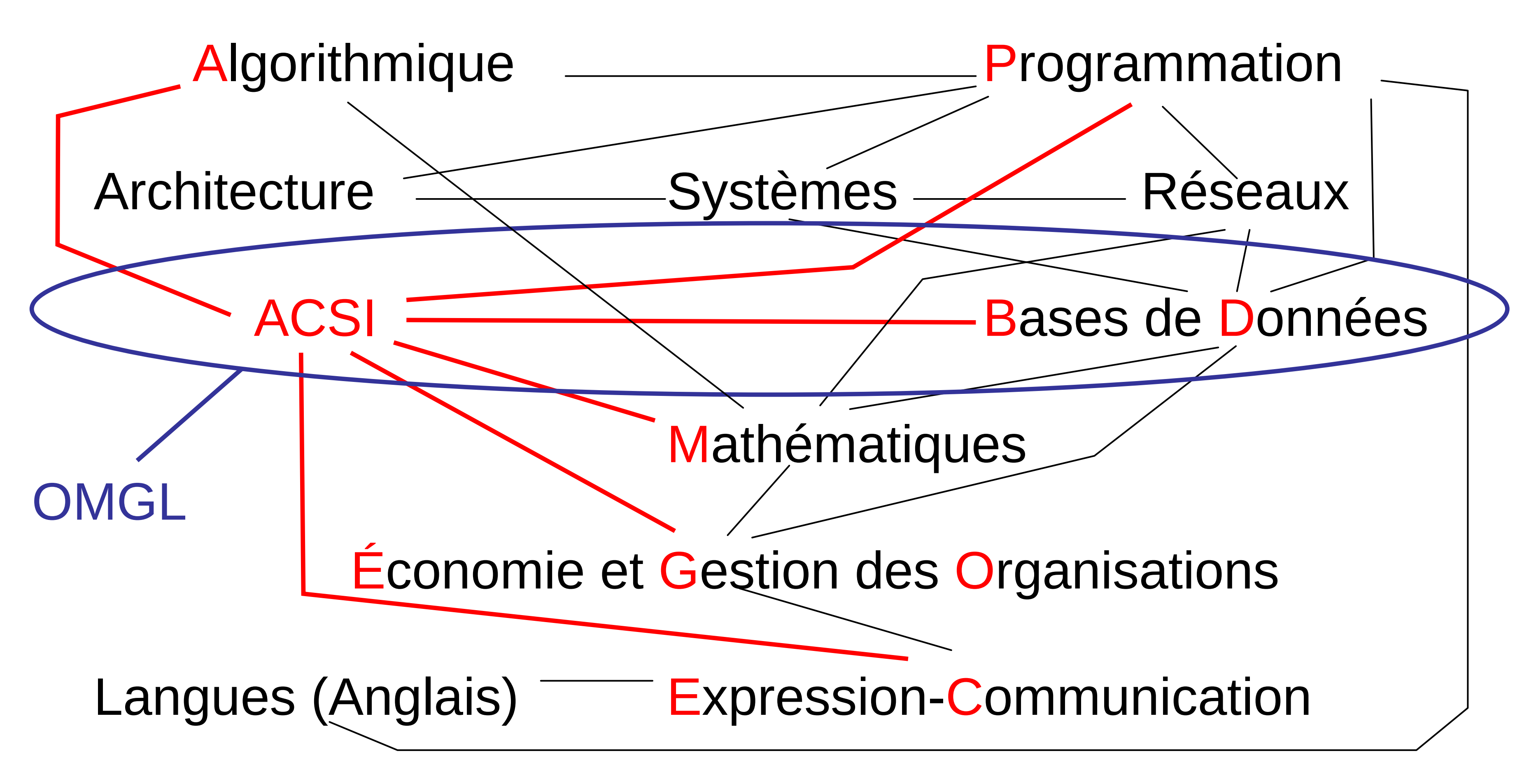
Cours d’Analyse et Conception des Systèmes d’Information (d’Outils et Modèles pour le Génie Logiciel)

7 novembre 2007

Olivier Guibert



Programme Pédagogique National

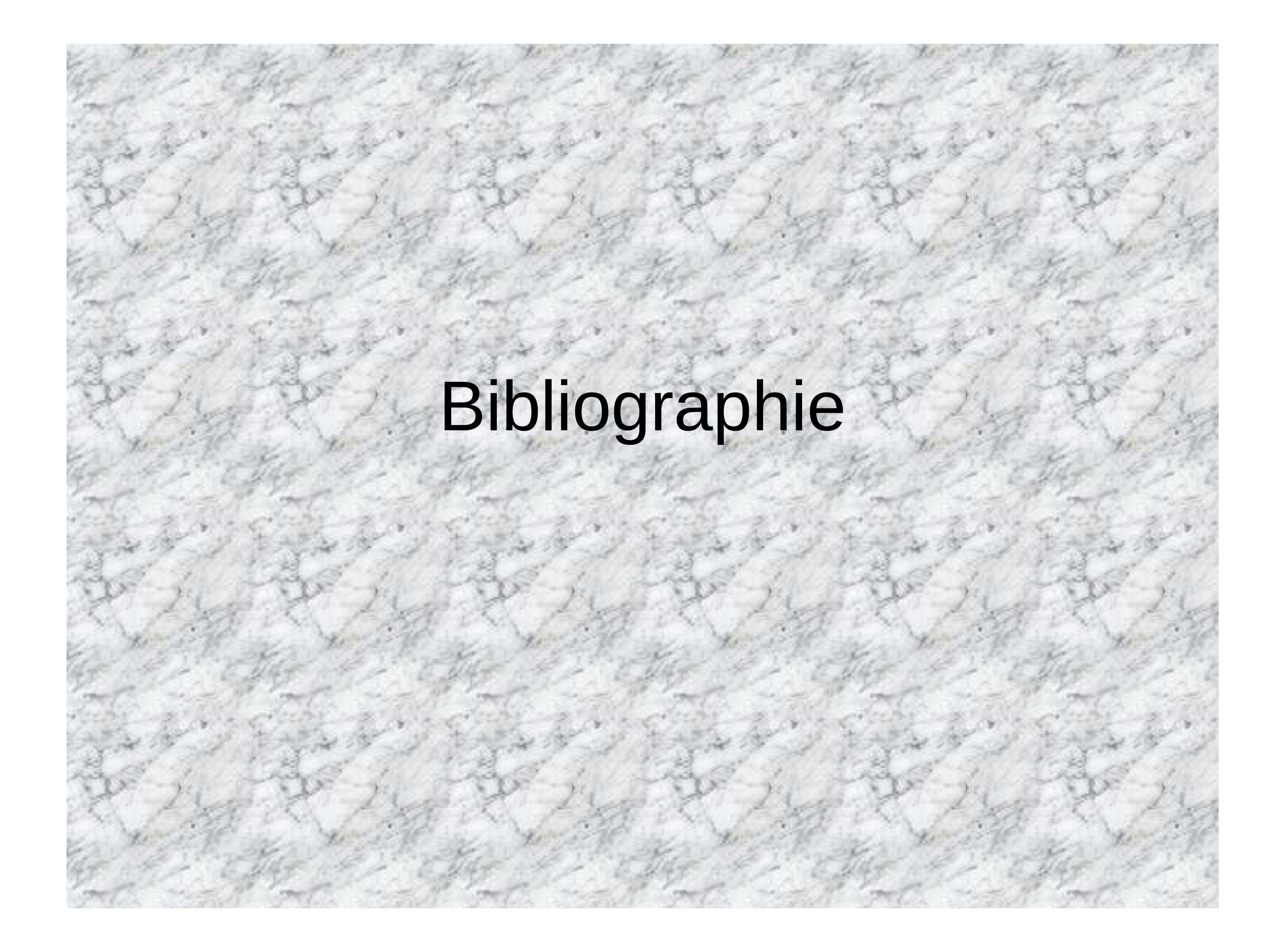


… et Projet Personnel et Professionnel, Projets Tutorés, Stage

Plan

- Bibliographie
- Définitions
- Intervenants
- Cycles de vie du logiciel
- Taxinomie des méthodes d'informatisation
- Démarche
- Modèles
- Méthodes et langages de modélisation

+ Gestion de Projet, Qualité, Interface Homme-Machine, etc.



Bibliographie

Bibliographie : génie logiciel

- ACSIOME, *Modélisation dans la conception des systèmes d'information*, Masson, 1989
- GALACSI, *Les systèmes d'information : analyse et conception*, Dunod, 1984
- GALACSI, *Comprendre les systèmes d'information : exercices corrigés d'analyse et de conception*, Dunod, 1985
- I. SOMMERVILLE, *Le génie logiciel et ses applications*, InterÉditions, 1985
- C. TESSIER, *La pratique des méthodes en informatique de gestion*, Les Editions d'Organisation, 1995
- P. ANDRÉ et A. VAILLY, *Conception des systèmes d'information – Panorama des méthodes et des techniques*, Ellipses, collection TECHNOSUP / Génie Logiciel, 2001
- P. ANDRÉ et A. VAILLY, *Spécification des logiciels – Deux exemples de pratiques récentes : Z et UML*, Ellipses, collection TECHNOSUP / Génie Logiciel, 2001

ACSIOME = Analyse et Conception des Systèmes d'Information : Outils, Modèles, Études
= M.C. HEYDEMANN, V. PRINCE, C. REYNAUD, F. SCHLIENGER et D. SCHLIENGER

GALACSI = Groupe d'Animation et de Liaison pour l'Analyse et Conception des Systèmes d'Information
= H. BRIAND, J.-B. CRAMPES, C. DUCATEAU, Y. HEBRAIL, D. HERIN-AIME, J. KOULOUMDJIAN et R. SABATIER

Bibliographie : systémique

- J-L. LE MOIGNE, *Les systèmes d'information dans les organisations*, Presses Universitaires de France, 1973
- J-L. LE MOIGNE, *Les systèmes de décision dans les organisations*, Presses Universitaires de France, 1974
- J-L. LE MOIGNE, *La théorie du système général - Théorie de la modélisation*, Presses Universitaires de France, 1977

- J. MÉLÈSE, *Analyse modulaire des systèmes*, Éditions Hommes et Techniques, 1977
- J. MÉLÈSE, *Approche systémique des organisations*, Éditions Hommes et Techniques, 1979

Bibliographie : bases de données

- M. ADIBA et C. DELOBEL, *Bases de données et systèmes relationnels*, Dunod, 1983

- G. GARDARIN, *Bases de données : les systèmes et leurs langages*, Eyrolles, 1983

Bibliographie : MERISE

- CGI (Compagnie Générale d'Informatique), *MERISE ou l'informatique avec méthode*, Nathan, 1986

- H. TARDIEU, A. ROCHFELD et R. COLLETTI, *La méthode MERISE, tome 1 : principes et outils*, Les Éditions d'Organisation, 1983
- H. TARDIEU, A. ROCHFELD, R. COLLETTI, G. PANET et G. VAHEE, *La méthode MERISE, tome 2 : démarches et pratiques*, Les Éditions d'Organisation, 1985
- A. ROCHFELD et J. MOREJON, *La méthode MERISE, tome 3 : gamme opératoire*, Les Éditions d'Organisation, 1989

- G. PANET et R. LETOUCHE, *MERISE/2, modèles et techniques MERISE avancés*, Les Éditions d'Organisation, 1994
- D. NANCI et B. ESPINASSE, Ingénierie des Systèmes d'Informations : *MERISE Deuxième génération*, Vuibert, 2001

Bibliographie : SADT

- D. T. ROSS, *Structured Analysis : A language for Communicating Ideas*, IEEE Transactions, Software engineering, vol. SE-3, n°1
- D. T. ROSS et K. E. SCHOMAN, *Structured Analysis for Requirements Definition*, IEEE Transactions, Software engineering , vol. SE-3, n°1
- IGL Technology, *SADT, un langage pour communiquer*, Eyrolles, 1989, 1993 (3ème édition)
- M. LISSANDRE, *Maîtriser SADT*, Colin, 1990
- P. JAULENT, *Génie logiciel : les méthodes*, Armand Colin, 1990

Bibliographie : méthodes objet

- S. SHLAER et S.J. MELLOR, *Object-Oriented Systems Analysis: Modeling the World in Data*, Englewood Cliffs, New Jersey: Yourdon Press, 1988
- S. BAILIN, *Remarks on Object-Oriented Requirements Specification*, Laurel, MD: Computer Technology Associates, 1988

- M. BOUZEGHOUB, G. GARDARIN et P. VALDURIEZ, *Les Objets : concepts, langages, bases de données, méthodes, interfaces*, Eyrolles, 1997

- J.-L. CAVARERO et R. LECAT, *La conception orientée objet, évidence ou fatalité*, Ellipses, collection TECHNOSUP / Génie Logiciel, 2000

Bibliographie : méthodes objet (« de MERISE à l'objet »)

- A. ROCHFELD et M. BOUZEGHOUB, *From Merise to OOM*, Revue Ingénierie des Systèmes d'Information, vol. 1, n°2, 1993
- J. MOREJON, *Merise : vers une modélisation objet*, Les Éditions d'Organisation, 1994
- M. BOUZEGHOUB, G. GARDARIN et P. VALDURIEZ, *Du C++ à Merise objet : Objets*, Eyrolles, 1994
- B. ESPINASSE, M. LAI et D. NANCI, *Merise+ : Une extension de la méthode Merise à l'approche objet par un apport de la méthode HOOD*, Revue Ingénierie des Systèmes d'Information, Hermès Éditeur, vol. 3, n°2-3, 1995
- B. ESPINASSE et D. NANCI, *Merise et l'approche orientée objet : du couplage avec OMT à une troisième génération*, Revue Ingénierie des Systèmes d'Information, Hermès Éditeur, vol. 5, n°4, 1997
- N. KETTANI, D. MIGNET, P. PARÉ et C. ROSENTHAL-SABROUX, *De Merise à UML*, Eyrolles, 1998
- J. GABAY, *Merise. Vers OMT et UML*, InterÉditions, 1998

Bibliographie : méthodes objet (OOA, Fusion, MCO, MACAO)

- P. COAD et E. YOURDON, *Object-oriented analysis*, Prentice Hall, 1990
- D. COLEMAN, P. ARNOLD, S. BODOFF, C. DOLLIN, H. GILCHRIST, J. HAYES et P. JEREMAES, *Fusion : la méthode orientée objet de 2ème génération*, Masson, 1992
- X. CASTELLANI, *MCO : Méthodologie d'analyse et de conception des systèmes à objets*, Masson, 1993
- J.-B. CRAMPES, *Méthode orientée-objet intégrale MACAO - Démarche participative pour l'analyse, la conception et la réalisation de logiciels*, Ellipses, collection TECHNOSUP / Génie Logiciel, 2003

Bibliographie : méthodes objet (HOOD, OMT, OOSE)

- G. BOOCH, *Object Oriented Design with Application*, The Benjamin/Cummings Publishing Company Inc., 1991
- G. BOOCH, *Analyse et Conception orientées objets*, Addison Wesley, 1994
- M. LAI, *Conception orientée objet, pratique de la méthode HOOD*, Dunod, 1991

- J. RUMBAUGH, M. BLAHA, W. PREMERLANI, F. EDDY et W. LORENSEN, *Object oriented modeling and design*, Prentice Hall, 1991

- I. JACOBSON, M. CHRISTERSON, P. JONSON et G. ÖVERGAARD, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison Wesley, 1992
- I. JACOBSON, *Le génie logiciel orienté objet*, Addison Wesley

Bibliographie : UML

- P.-A. MULLER, *Modélisation objet avec UML*, Eyrolles, 1998
- C. MORLEY, B. LEBLANC et J. HUGUES, *UML pour l'analyse d'un système d'information – Le cahier des charges du maître d'ouvrage*, Dunod, 2000
- C. SOUTOU, *Objet-Relationnel sous Oracle8, Modélisation avec UML*, Eyrolles, 1999
- P. ROQUES et F. VALLÉE, *UML en action – De l'analyse des besoins à la conception en Java*, Eyrolles, 2000
- M. LAI, *Penser objet avec UML et Java*, InterÉditions, 1998
- M. LAI, *UML : La notation unifiée de modélisation objet – De Java aux EJB*, Dunod, 2000

- G. BOOCH, J. RUMBAUGH et I. JACOBSON, *The Unified Modeling Language User Guide*, Addison-Wesley, 1999
- I. JACOBSON, G. BOOCH et J. RUMBAUGH, *The Unified Software Development Process*, Addison-Wesley, 1999
- J. RUMBAUGH, I. JACOBSON et G. BOOCH, *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1999

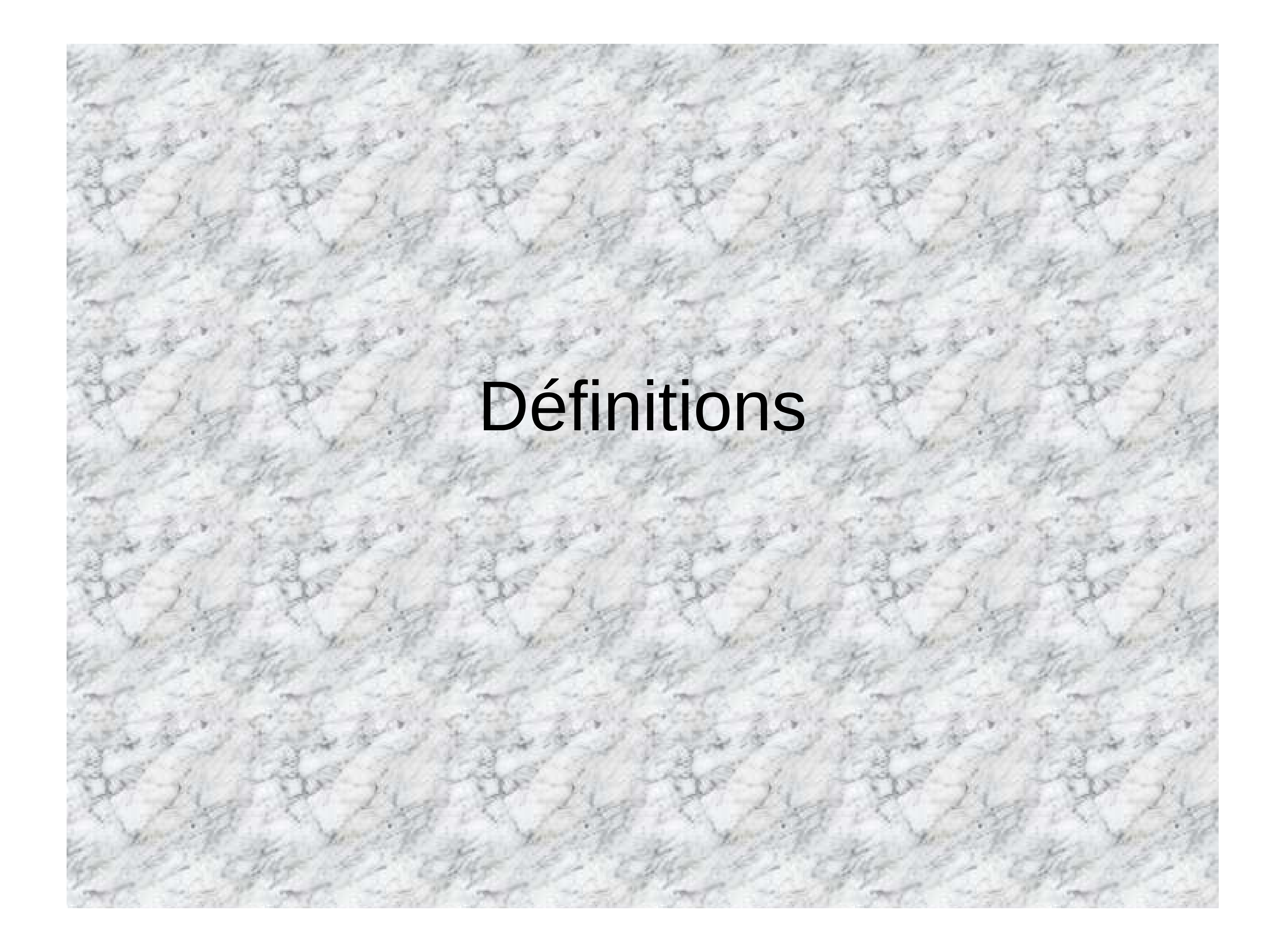
Bibliographie : B

- M. SPIVEY, *La notation Z*, Masson - Prentice Hall, 1992
- D. LIGHTFOOT, *Spécification formelle avec Z*, TEKNEA, 1994

- J.-R. ABRIAL, *The B-book: Assigning Programs to Meanings*, Cambridge University Press, 1996
- J.-R. ABRIAL, *Introduction à la méthode B*, 6 vidéo-cassettes, IUT de Nantes
- J.-R. ABRIAL, *La méthode B - études de cas*, 6 vidéo-cassettes, IUT de Nantes
- H. HABRIAS, *Introduction à la spécification*, Masson, 1993
- H. HABRIAS, *Spécification formelle avec B*, Éditions Hermès – Lavoisier, 2001

Bibliographie : interface homme-machine

- J.-B. CRAMPES, *Interfaces graphiques ergonomiques - Conception et Modélisation*, Ellipses, collection TECHNOSUP / Génie Logiciel, 1997



Définitions

Définitions : OMGL

- OMGL = Outils et Modèles pour le Génie Logiciel

- **Outil** : logiciel supportant une méthode
- **Modèle** : représentation schématique de la réalité
- **Logiciel** *selon l'arrêté du 22 décembre 1981* : ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitements de l'information
- **Génie Logiciel** (ou l'ingénierie des systèmes d'information) *selon l'arrêté du 30 décembre 1983* : ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et de son suivi

Définitions : ACSI

- ACSI = Analyse et Conception des Systèmes d'Information

- **Analyse** : processus d'examen de l'existant
- **Conception** : processus de définition de la future application informatique
- **Systèmes d'Information** : ensemble des moyens (humains et matériels) et des méthodes se rapportant au traitement de l'information d'une organisation

Définitions : BD

- BD = Bases de Données

- **Bases de Données** *[définition des informaticiens]* : ensemble des données (de l'organisation) structurées et liées entre elles :
  – stocké sur support à accès direct (disque magnétique)
  – géré par un SGBD (Système de Gestion de Bases de Données)
  – accessible par un ensemble d'applications

Définitions (compléments)

- Informatique : science du traitement automatique et rationnel de l'information [académie française, 1966]

- Informatique de Gestion : informatisation des systèmes d'information

- AGL = Atelier de Génie Logiciel (*CASE* = *Computer Aided Software Engineering*) : ingénierie du logiciel assisté par ordinateur

L’information, indispensable dans le processus de décision d'une organisation

- Diminution de l'incertitude
- Liberté de choix
- Cohésion de l'organisation
- Évolutivité par rapport à l'environnement

Qualités requises pour une information

- Pertinence (mesure la qualité d'une information) : relation directe entre l'action à accomplir ou la décision à prendre
  - précision : ni trop importante, ni trop faible
  - sécurité (pour reconstituer l'information en cas d'accident)
  - intégrité (contraintes statiques ou dynamiques)
  - confidentialité (protection contre tentatives d'accès)
  - non redondance (un seul exemplaire de chaque information)
  - Convivialité (qualité de représentation sur support externe et facilité d'accès par les utilisateurs)
  - âge (temps entre enregistrement et sortie des résultats)
  - fréquence (nombre de transmissions par unité de temps)
- Cohérence (d'unité, de temps, etc.)
- Rentabilité : coût d'obtention ≤ gain, meilleur service

Types d'information

- Niveau d'agrégation
  - brutes
  - élaborées
- Flux
  - logistique
  - monétaire
  - de personnel
  - de l'actif
- Utilisation
  - planification stratégique
  - gestion administrative
  - régulation opérationnelle
- Nature du support
  - oral
  - documentaire
  - informatique

Définitions : systémique

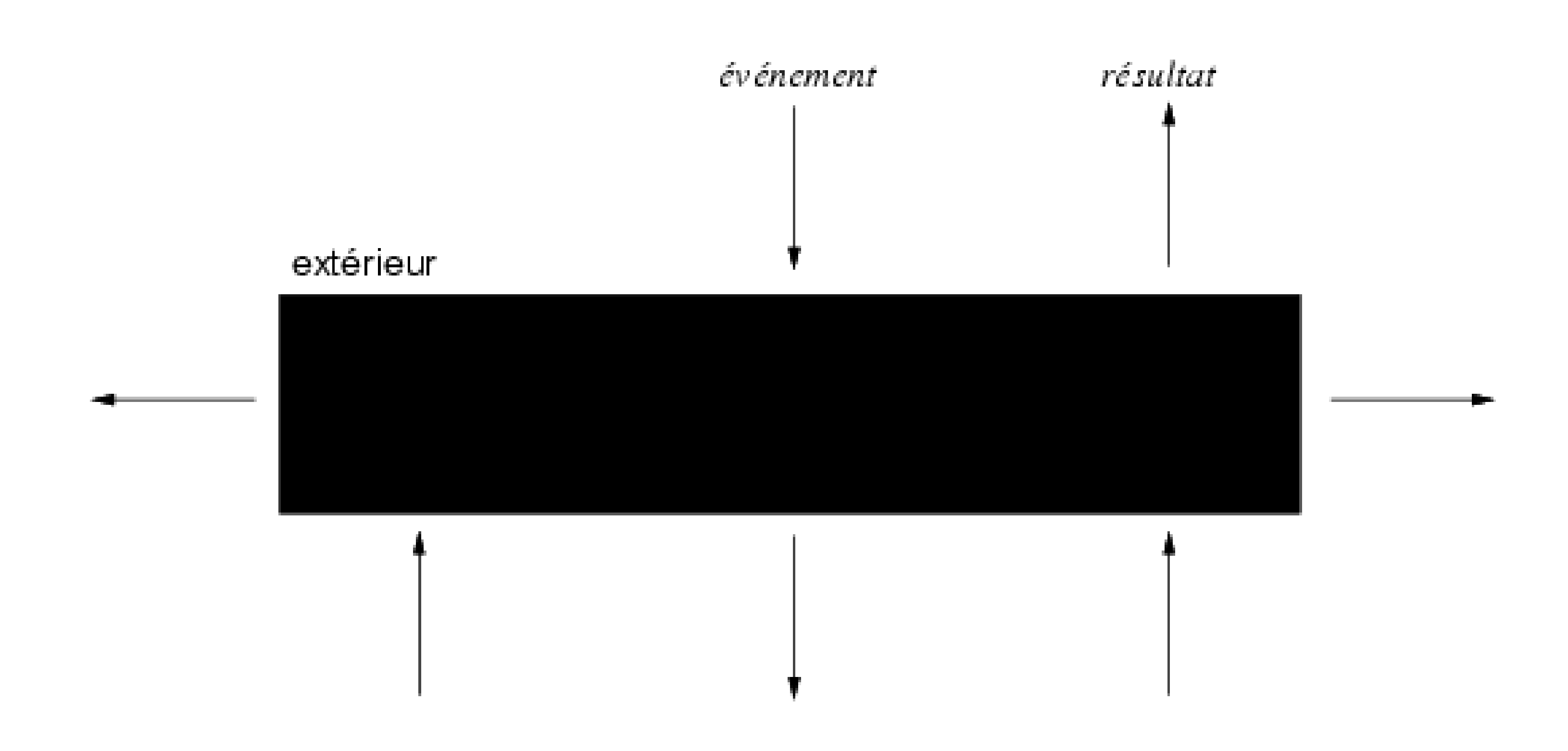
- Analyse systémique : analyse qui envisage les éléments d'une conformation complexe, les faits (notamment les faits économiques), non pas isolément mais globalement, en tant que parties intégrante d'un ensemble dont les différents composants sont dans une relation de dépendance réciproque [P.L.I. 2003]

- Neuf niveaux imbriqués de complexité selon cette théorie : l'objet passif, l'objet actif, l'objet actif régulé, l'objet s'informe, l'objet décide son activité, l'objet actif a une mémoire, l'objet actif se coordonne, l'objet actif imagine (et donc s'auto-organise), l'objet actif s'auto-finalise
  L'organisation correspond au dernier niveau

Définitions : système

- Système : ensemble d'éléments en interaction dynamique, dont les éléments sont organisés et coordonnés en vue d'atteindre un objectif, qui évolue dans un environnement

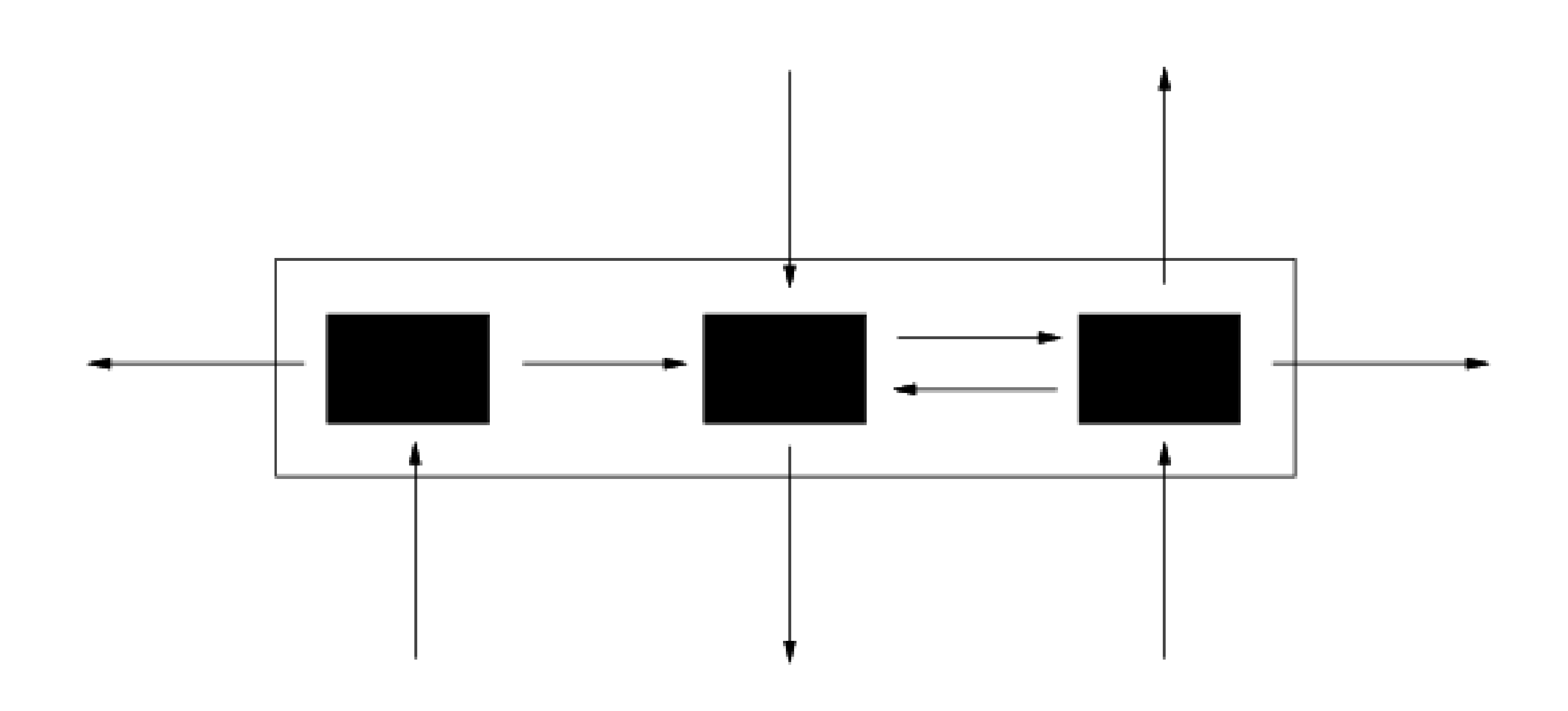
Un système vu comme une « boîte noire »



Système : de la « boîte noire » à la « boîte blanche »

Le système se décompose en sous-systèmes dont on définit les entrées (issues de l'extérieur ou sorties d'autres sous-systèmes) et les sorties (à destination de l'extérieur ou devenant les entrées d'autres sous-systèmes)

Système : de la « boîte noire » à la « boîte blanche »



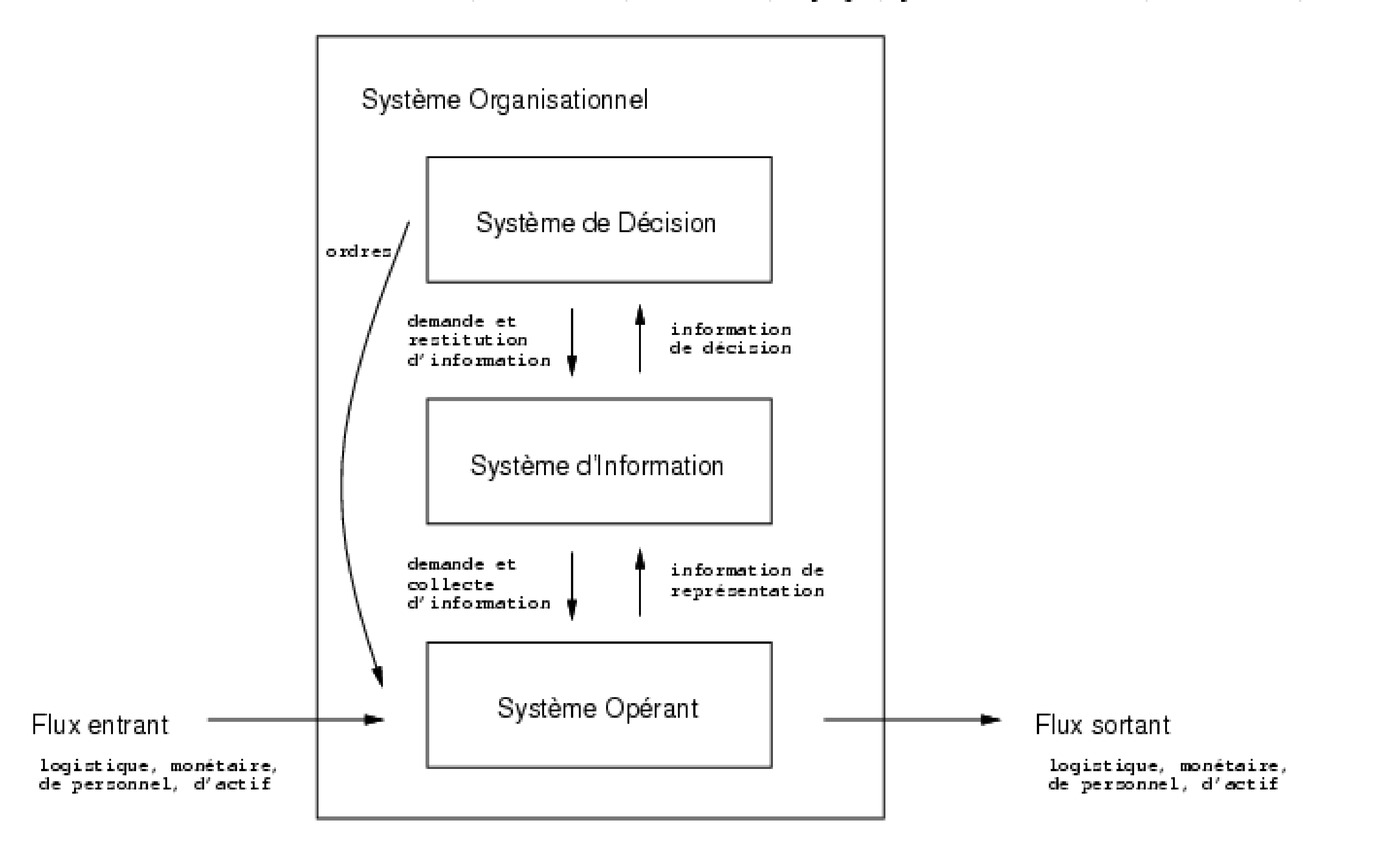
Système : de la « boîte noire » à la « boîte blanche »

Chaque sous-système est lui-même un système : affinages successifs jusqu'à l'obtention d'une « boîte blanche »

Principales difficultés de l'approche d'un système par décomposition récursive

- identification du système
- identification des limites du système
- identification des sous-systèmes
- risque de perte engendrée par la décomposition
- etc.

Définitions : système organisationnel



Définitions : système organisationnel

- Système de Décision (ou pilotage, management, etc.)
  - Guide l'organisation vers ses objectifs (activités de planification et de contrôle) : coordonne, imagine, finalise, élabore objectifs
  - *Gérer*
- Système d'Information
  - Intermédiaire entre les systèmes de décision et opérationnel, par qui transite toute information :
    - mémorise l’information (conservation de l'information pour des besoins ultérieurs),
    - traite l’information (rapprochements, calculs, comparaisons),
    - fait circuler l’information (accès à la mémoire, échange entre acteurs)
- Système Opérant (ou logistique, technologique, physique, de production, etc.)
  - Effectue la transformation : reçoit, traite, envoie
  - *Acheter* ; *Produire* ; *Stocker* ; *Vendre*

<u>Remarque</u> : un même employé peut être un acteur de chacun des trois sous-systèmes

Rôles du système d’information

- Produire les informations légales réclamées par l'environnement
- Déclencher les décisions programmées
- Fournir des informations aux décideurs pour aider à la prise de décisions non programmées
- Coordonner les tâches en assurant les communications au sein du système organisationnel

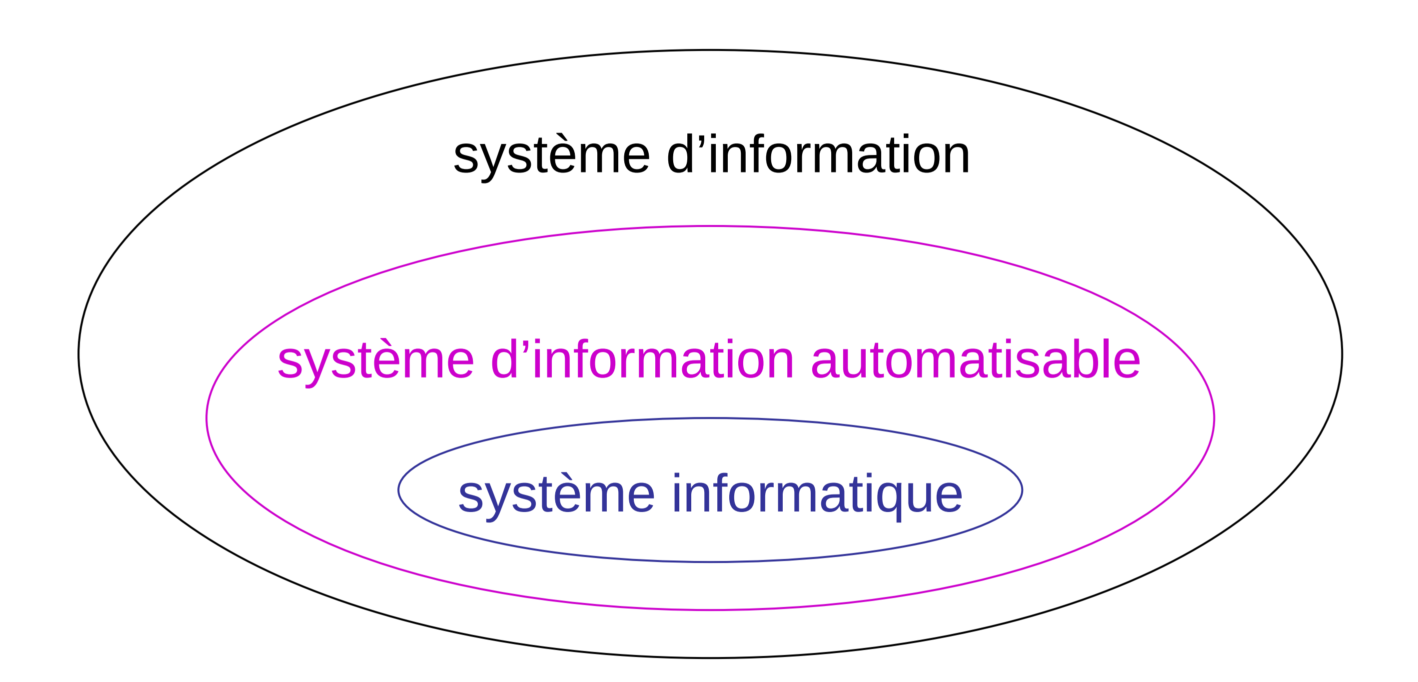
Connaissances nécessaires en Informatique de Gestion

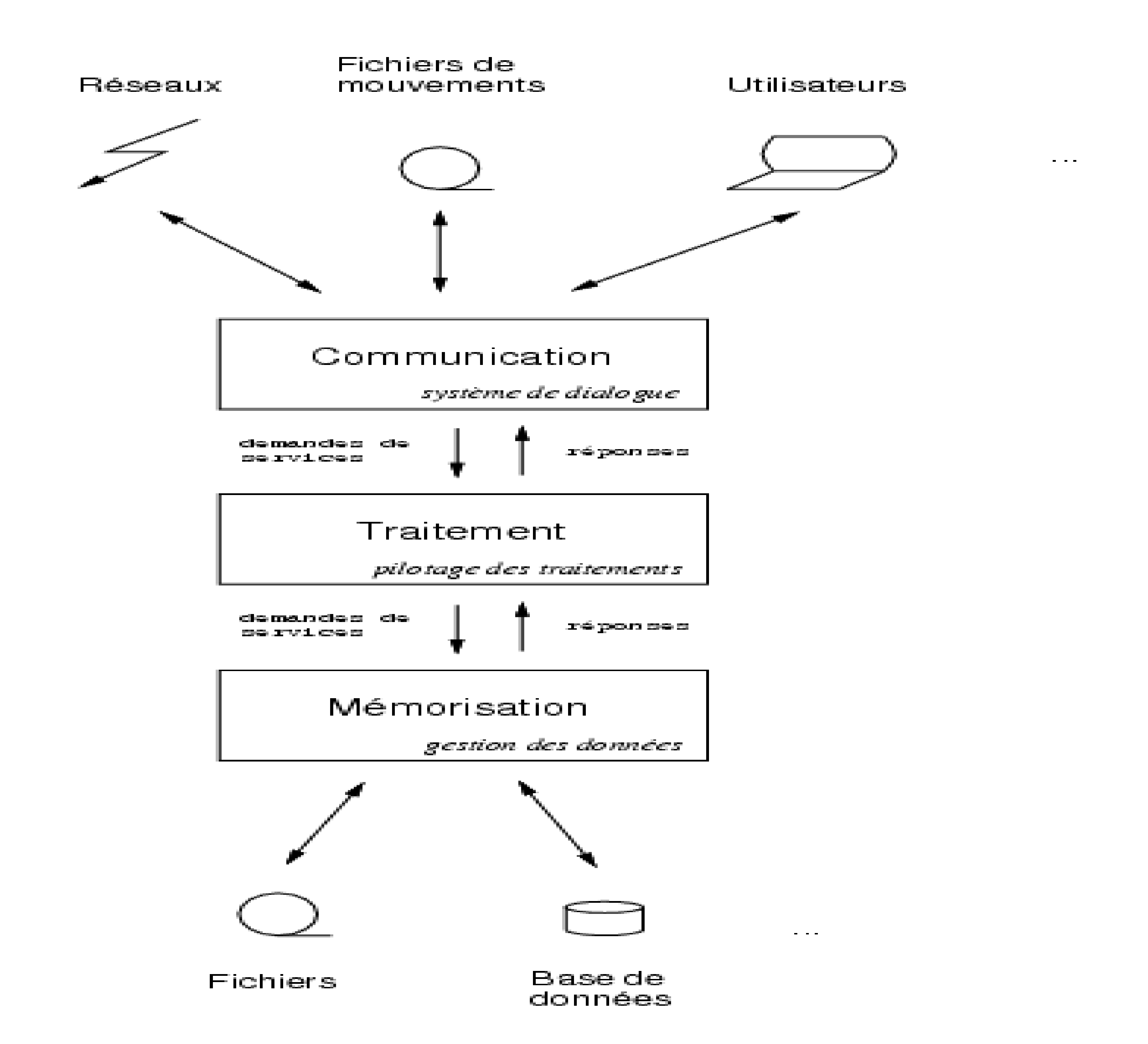
- Science de gestion : mise en place du réseau d'information et de communication (conception du système d'information)

- Technique informatique : conception et réalisation du système informatique pour gérer le système d'information (conception du logiciel)

Définitions : système d'information vs système informatique

- Le système informatique est la partie informatisée du système d'information automatisable




Réseaux
Fichiers de mouvements
Utilisateurs
...
Communication
système de dialogue
demandes de services
réponses
Traitement
pilotage des traitements
demandes de services
réponses
Mémorisation
gestion des données
...
Fichiers
Base de données

Définitions : système informatique

- Communication
  - Système informatique communique directement avec son environnement (utilisateurs, fichiers d'autres systèmes via un réseau ou non, etc.)
  - Communication entre composants d'une application (ex. : fichier de mouvement)
- Traitement
  - Demandes de traitements issues de l'échange entre le système informatique et son environnement
  - Pilotage des traitements proposés par le système informatique en gérant les appels aux processus permettant de les réaliser
- Mémorisation
  - Gestion des données par différents modes d'accès (et stockage aux niveaux logique et physique)

Enjeux de l’informatisation pour l'organisation

- Augmenter la productivité en améliorant l’efficacité des utilisateurs
- Améliorer les conditions de travail : enrichissement des tâches
- Rendre un meilleur service (de qualité, rapide, etc.) aux partenaires de l'organisation

Facteurs de la complexité de l'informatisation

- Difficultés techniques de l'informatique : complexité de la mise en œuvre des matériels, complexité de la construction logicielle, réflexion abstraite, contraintes techniques
- Constantes novations (matérielle et logicielle)
- Symbiose requise entre l'application informatique et toute l'organisation (et ses partenaires)
- Multiplicité des décisions et nombreux domaines (humain, financier, technique, etc.) de l'organisation concernés

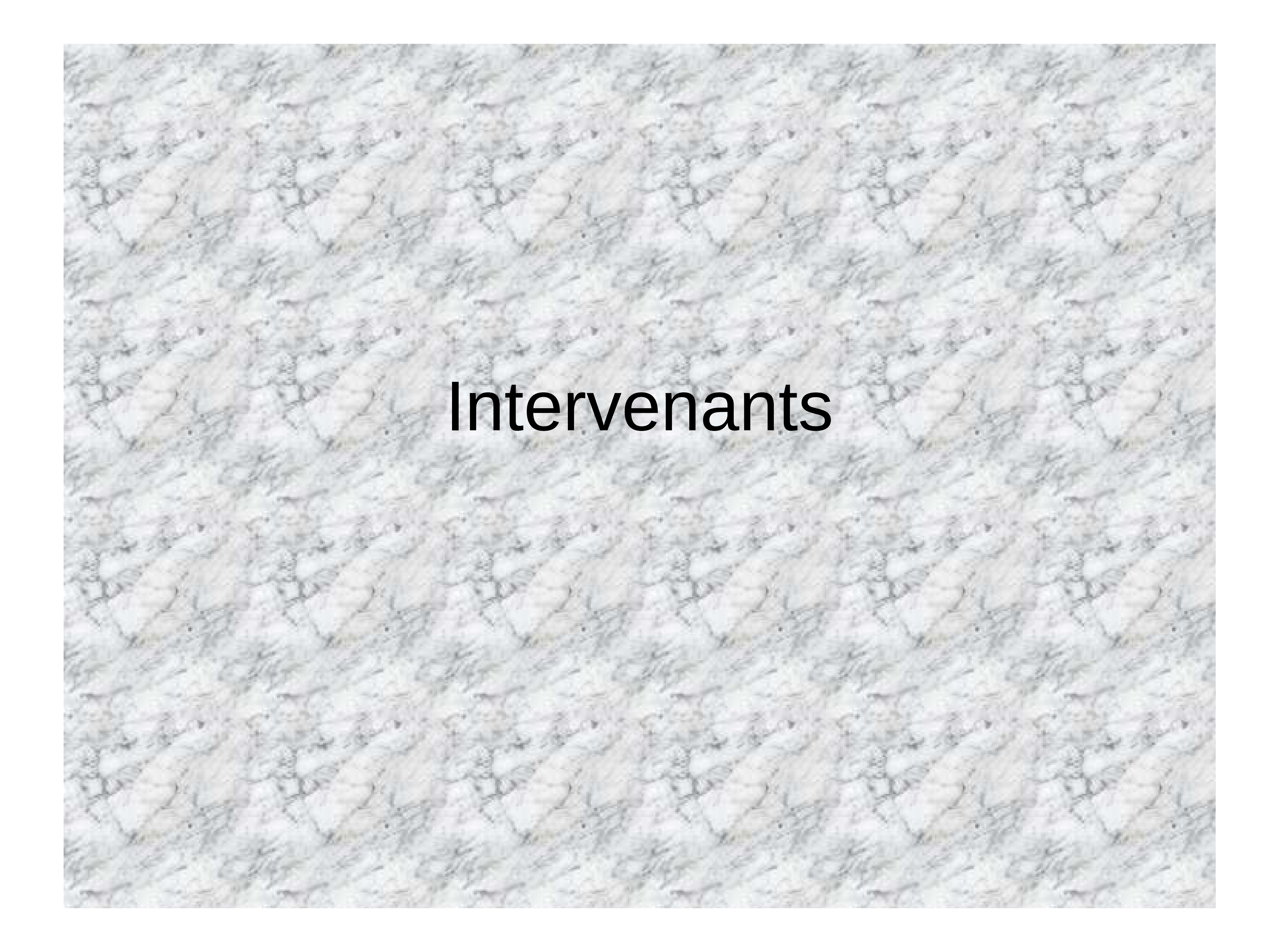
Critères d'un bon système informatique

- Productivité (en rationalisant le processus d'informatisation)
  - Établissement d'une ligne directrice des informatisations
  - Planification et suivi des performances
  - Efficacité des études informatiques
  - Utilisation judicieuse des technologies
- Qualité
  - Conformité de la réalisation par rapport aux besoins
  - Documentation correcte
  - Adaptabilité
  - Fiabilité
  - Facilité d'utilisation
- Rentabilité (i.e. gain pour l'organisation relativement au coût de l'informatisation)

L'informatique remplit maintenant un rôle stratégique dans l'organisation

On est passé de l’automatisation des tâches administratives aux systèmes d'information d'aide à la décision (SIAD)

Informatique de production →	Informatique de management
• Système opérant	• Système décisionnel
• Début années 1960	• Plus récent
• Faible complexité des traitements	• Forte complexité des traitements
• Mise à jour transactionnelles, chaînes séquentielles	• Consultation en temps partagé
• Information précise	• Information significative, rapidement disponible
• L3G	• SQL


Intervenants

Intervenants : les départements du service informatique (01 Informatique 27/10/1995)

- Direction informatique

  Responsable du service informatique ; Chef d'un département du service informatique

- Expertise

  Administrateur ou expert en système (d'exploitation), réseau, base de données, méthodes, qualité, sécurité, technologies diverses

- Études - Développement

  Chef de projet ; Analyste ; Concepteur ; Développeur (ou programmeur)

- Production - Exploitation

  Opérateur - Pupitreur ; Analyste d'exploitation ; Contrôleur réseau ; Technicien (micro-informatique, réseau, messagerie, téléphonie)

- Support et assistance

  Assistant technique clientèle

Autre métier : Consultant en systèmes d'information

Intervenants : anciens diplômés du département informatique de l’IUT de l’université Bordeaux 1

(statistiques élaborées à partir des 530 réponses reçues sur 2156 diplômés au 18/12/1996)

- Direction informatique ……………………………….. 16 %
- Expertise ………………………………………………. 10 %
- Études – Développement …………………………… 54 %
- Production - Exploitation ; Support et assistance ... 12 %
- *Non informaticien* ……………………………………..... 8 %

Intervenants : MOA vs MOE

- La maîtrise d'ouvrage (MOA) : les utilisateurs
  - Direction générale
  - Responsable du service des utilisateurs
  - Personnel
  - Autres services
  - Clients
- La maîtrise d'œuvre (MOE) : les informaticiens, prestataires de services
  - Responsable du service informatique
  - Chef de projet
  - Analyste
  - Développeur
  - Personnel de l’exploitation
  - Sous-traitants de l'application

Nomenclature 2005 des emplois-métiers

Les emplois-métiers du système d’information dans les grandes entreprises

CIGREF (club informatique des grandes entreprises françaises)

février 2005

http://www.cigref.fr/cigref/livelink.exe/Nomenclature_RH_2005.pdf

Nomenclature 2005 : 6 familles

- Conseil en système d'information et maîtrise d'ouvrage (6 métiers)
- Support et assistance aux utilisateurs (3 métiers)
- Production et exploitation (7 métiers)
- Études, développement et intégration (4 métiers)
- Support et assistance technique interne (6 métiers)
- Administration et gestion de la direction du système d'information (5 métiers)

Nomenclature 2005 : 31 métiers

- Conseil en système d'information et maîtrise d'ouvrage
  - Consultant en systèmes d’information
  - Urbaniste des systèmes d’information
  - Chef de projet maîtrise d’ouvrage
  - Responsable du système d’information « métier »
  - Gestionnaire d’applications
  - Responsable de projet « métier »
- Support et assistance aux utilisateurs
  - Assistant fonctionnel
  - Technicien support SVP
  - Chargé d’affaires internes

Nomenclature 2005 : 31 métiers

- Production et exploitation
  - Technicien d'exploitation
  - Technicien poste de travail
  - Technicien réseaux ou télécoms
  - Administrateur d'outils / systèmes / réseaux et télécoms
  - Administrateur de bases de données
  - Intégrateur d'exploitation
  - Pilote d'exploitation
- Études, développement et intégration
  - Chef de projet maîtrise d'œuvre
  - Développeur
  - Intégrateur d'applications
  - Paramétreur de progiciels de gestion intégré (PGI i.e. *ERP, enterprise resource planning*)

Nomenclature 2005 : 31 métiers

- Support et assistance technique interne
  - Expert système d’exploitation
  - Expert réseaux / télécoms
  - Expert méthode et outils / qualité / sécurité
  - Expert en technologie internet / intranet et multimédia
  - Responsable sécurité des systèmes d’information
  - Architecte technique
- Administration et gestion de la DSI
  - Responsable du management de la DSI
  - Responsable d’exploitation informatique
  - Responsable d’une entité informatique
  - Responsable de(s) service(s) administratif(s) et financier(s) de la DSI
  - Responsable Télécoms

Nomenclature 2005 : développeur

- Synonymes
  - Analyste-programmeur
  - Réalisateur en informatique
  - Analyste fonctionnel
  - Analyste réalisateur

Nomenclature 2005 : développeur

- Mission

  À la demande de la maîtrise d’œuvre, et sur la base des spécifications fonctionnelles émises par celle-ci, le développeur analyse, paramètre et code les composants logiciels applicatifs dans le respect des normes et procédures, ainsi que les évolutions souhaitées

Nomenclature 2005 : développeur

- Activités et tâches
  - Analyse

    Définition de spécifications ; Analyse organique ; Adaptation et paramétrage de progiciels applicatifs ; Prototypage
  - Développement

    Réalisation de modules (objets et composants logiciels) ; Assemblage de ces éléments ; Rédaction de documentations ; Industrialisation de composants et d'applications
  - Qualification

    Élaboration de jeux d'essais (tests unitaires d'intégration) ; Tests ; Identification et traitement des dysfonctionnements
  - Maintenance

    Maintenance corrective ; Maintenance évolutive ; Administration des composants logiciels réutilisables et gestion de la nomenclature de ces composants

Nomenclature 2005 : développeur

- Parcours professionnel
  - Profil : Bac + 2 ou 3
  - Expérience : Débutant

Nomenclature 2005 : développeur

- Tendances et facteurs d’évolution
  - Usage croissant des progiciels, d’où importance croissante du paramétrage, de l’objet, du fonctionnel aux dépens du développement spécifique, de l’algorithmique
  - Renouvellement rapide des langages : java, langages objet…
  - Importance croissante de l’ergonomie
  - Durée de vie des applications raccourcie
  - Souci de réutilisation des développements

Nomenclature 2005 : développeur

- Savoir-faire système d'information
  - Expertise
    - Langages de programmation [Développement]
    - Méthodes, normes et outils de développement [Développement]
  - Maîtrise
    - Conception, modélisation et architecture d'applications [Conception]
    - Algorithmique [Développement]
    - Techniques de développement (maquettage et prototypage, client-serveur, objet, RAD) [Développement]
    - Charte d'utilisation et de sécurité des SI [Sécurité informatique]
  - …/…

Nomenclature 2005 : développeur

- Application
  - Parc applicatif et de services [Architecture applicative / fonctionnelle]
  - Paramétrage d'applications [Développement]
  - Intégration de logiciels [Intégration]
  - Intégration de matériels [Intégration]
  - Gestion de production [Production - Exploitation]
  - Normes et procédures de sécurité I&T (Informatique et Télécoms) [Sécurité informatique]
- Notions
  - Architecture de systèmes d'exploitation [Architecture technique]
  - Administration de bases de données [Gestion de données - Bases de données]
  - Intégration de systèmes d'exploitation [Intégration]
  - Environnements d'exploitation [Production - Exploitation]
  - Logiciels et matériels réseaux [Télécom - Réseaux]

Nomenclature 2005 : développeur

- Savoir-faire généraux
  - Expertise
  - Maîtrise
    - Ergonomie et interfaces homme-machine [Savoirs de base]
  - Application
    - Compréhension des clients de la DSI (utilisateurs fonctionnels) et de leurs besoins [Connaissances des métiers de l'entreprise]
    - Techniques de l'assurance qualité [Qualité]
    - Capacité rédactionnelle [Savoirs de base]
  - Notions
    - Culture générale I&T [Connaissances des métiers de l'entreprise]
    - Pratique de l'anglais technique lu, écrit et parlé [Langue]

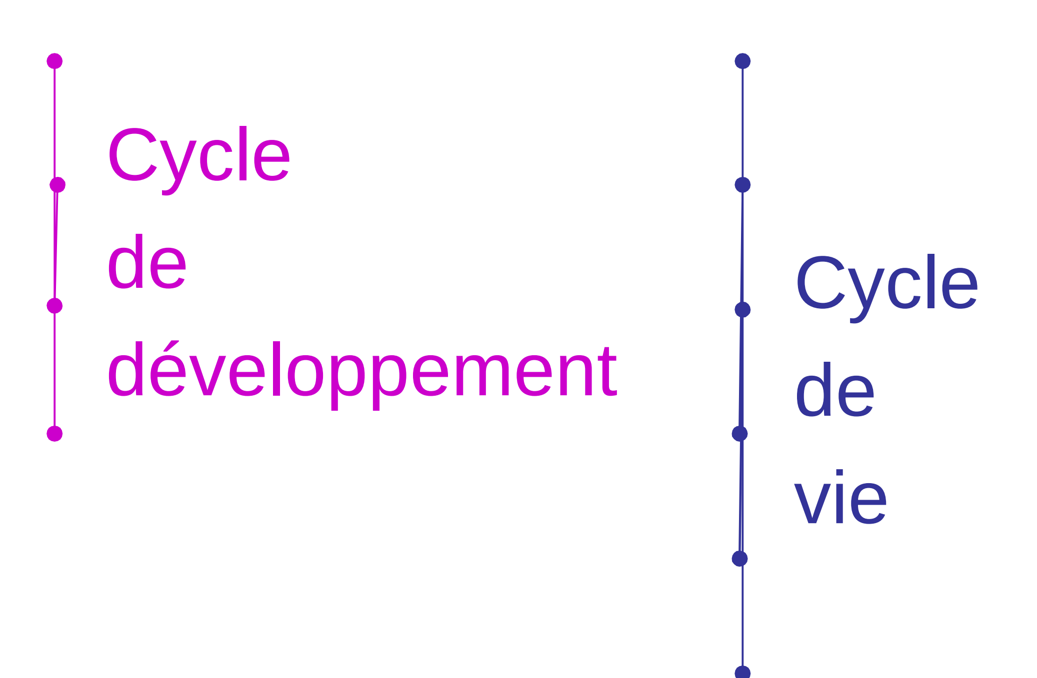
Nomenclature 2005 : développeur

- Aptitudes comportementales
  - Essentiel
    - Méthode [Compétences de résolution de problèmes]
    - Analyse [Compétences de résolution de problèmes]
    - Rigueur [Compétences d'efficacité personnelle]
  - Utile
    - Logique [Compétences de résolution de problèmes]
    - Adaptabilité [Compétences d'efficacité personnelle]
    - Gestion de situation [Compétences d'efficacité personnelle]
    - Pragmatisme [Compétences d'efficacité personnelle]
    - Écoute et communication [Compétences relationnelles]
    - Travail en équipe [Compétences relationnelles]


Cycles de vie du logiciel

Cycle de développement et cycle de vie du logiciel : les phases

- Analyse
- Conception
- Réalisation
- Tests
- Exploitation
- Maintenance



Cycles de vie du logiciel

- Analyse de l'existant et définition des besoins, du système d'information et du logiciel

- Conception du système d'information et du logiciel

- Réalisation (ou codage, programmation) : traduction des algorithmes dans un langage compréhensible par un ordinateur

Cycles de vie du logiciel

- Tests :
  - vérification du logiciel (i.e. système informatique)
  - validation du logiciel
  - vérification du système d'information
  - validation du système d'information

Vérification : le produit en cours d'élaboration répond-il à la définition des besoins ? (est-ce bien le produit ?)

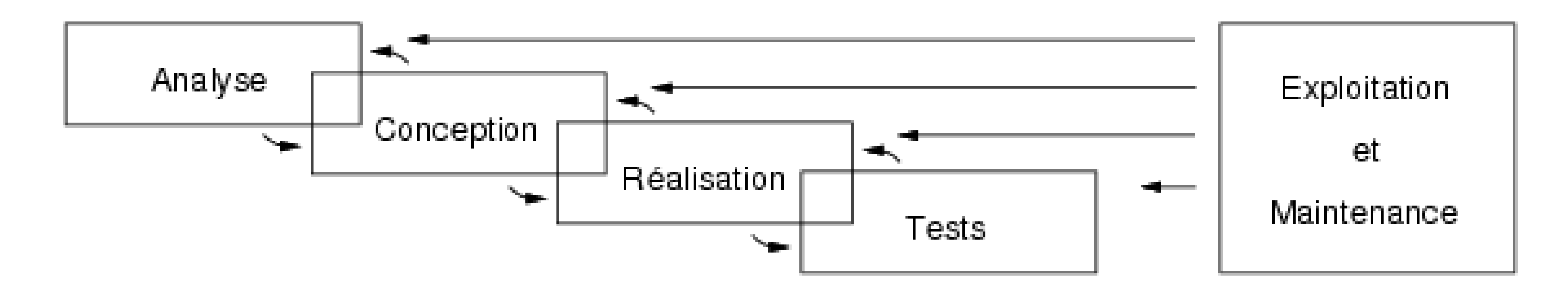
Validation : le produit en cours d'élaboration remplit-il les fonctionnalités désirées par l'utilisateur ? (est-ce le bon produit ?)

Cycles de vie du logiciel

- Exploitation : utilisation du logiciel une fois installé (et dont on fait la recette)

- Maintenance
  - Correction des erreurs
  - Amélioration des fonctions existantes
  - Ajout de nouvelles fonctionnalités

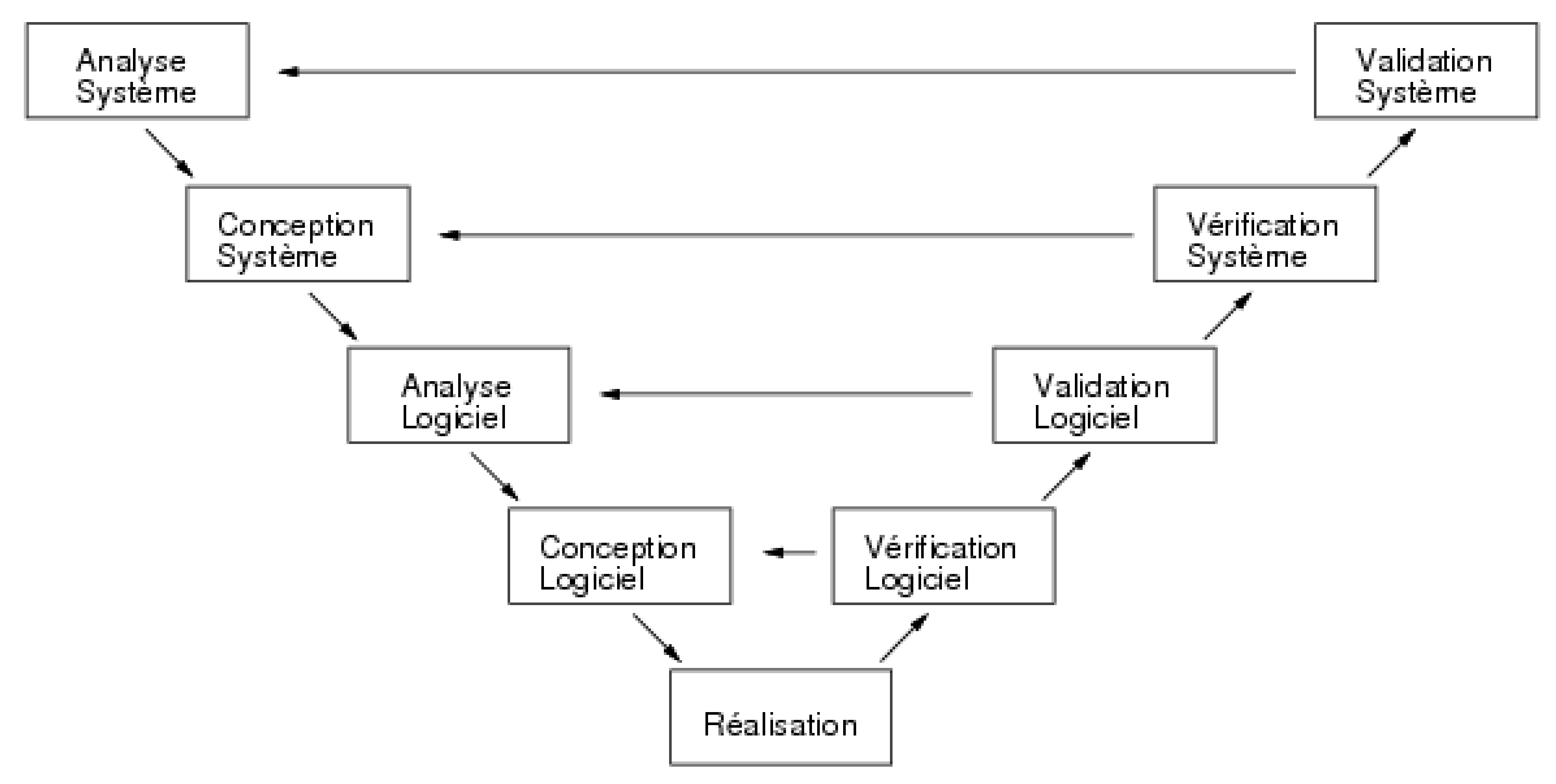
Cycles de vie en cascade (ou en chute d'eau)



Critiques :

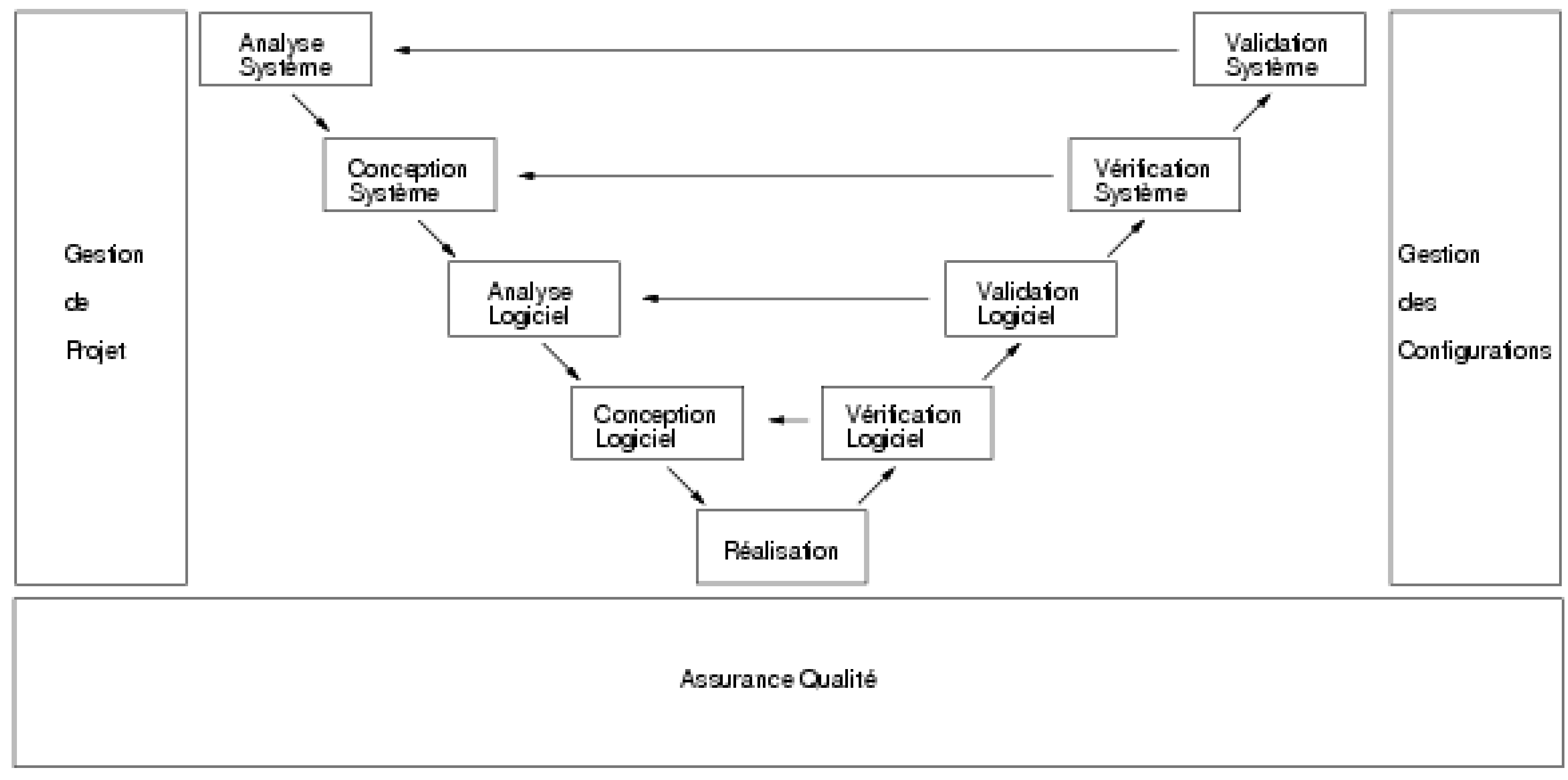
- Recouvrement de phases
- Avancées et retours d'une seule phase du cycle de développement à la fois
- Impact de la maintenance sur toutes les phases du développement
- Contacts avec l'utilisateur restreints à la phase d'analyse

Cycles de développement en V



- Système signifie ici système d'information (manuel et informatisé)
- Modèle de l'AFCIQ (Association Française pour le Contrôle Industriel de Qualité) avec le vocabulaire suivant : Spécification fonctionnelle \ Conception préliminaire \ Conception détaillée \ Codage / Tests unitaires / Tests d'intégration / Recette

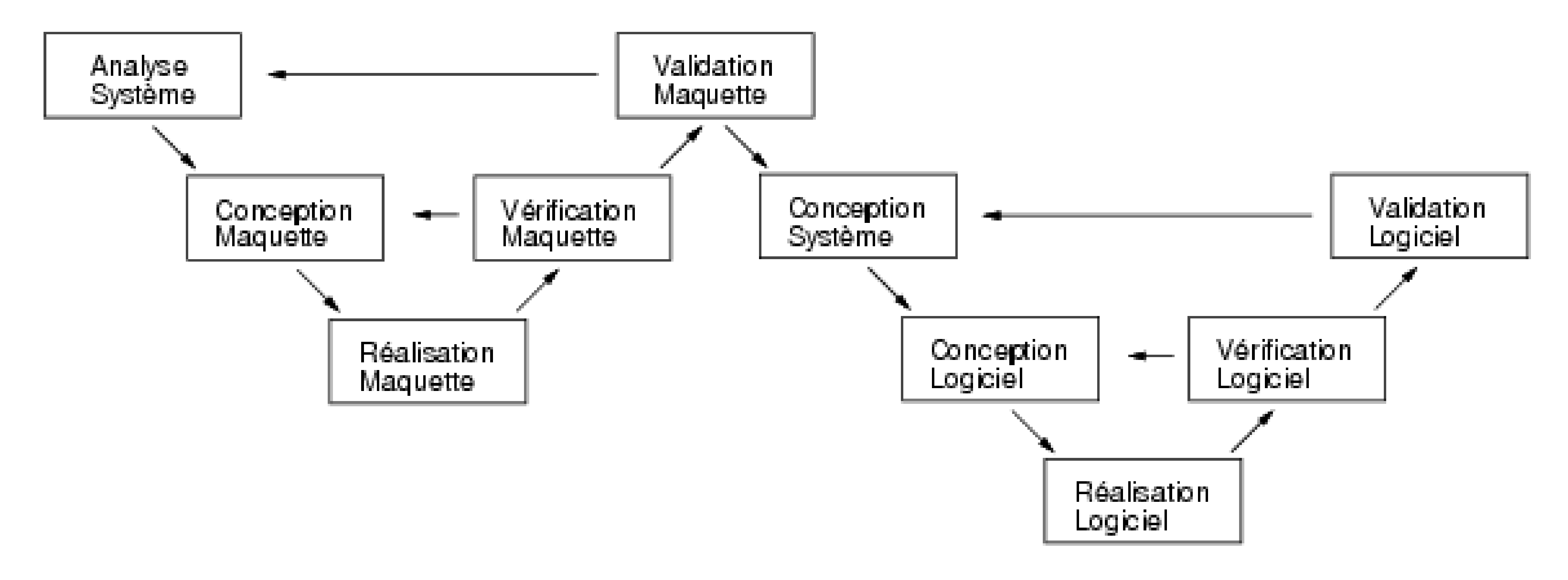
Cycles de développement en M



3 activités interviennent durant toute la durée du développement en V

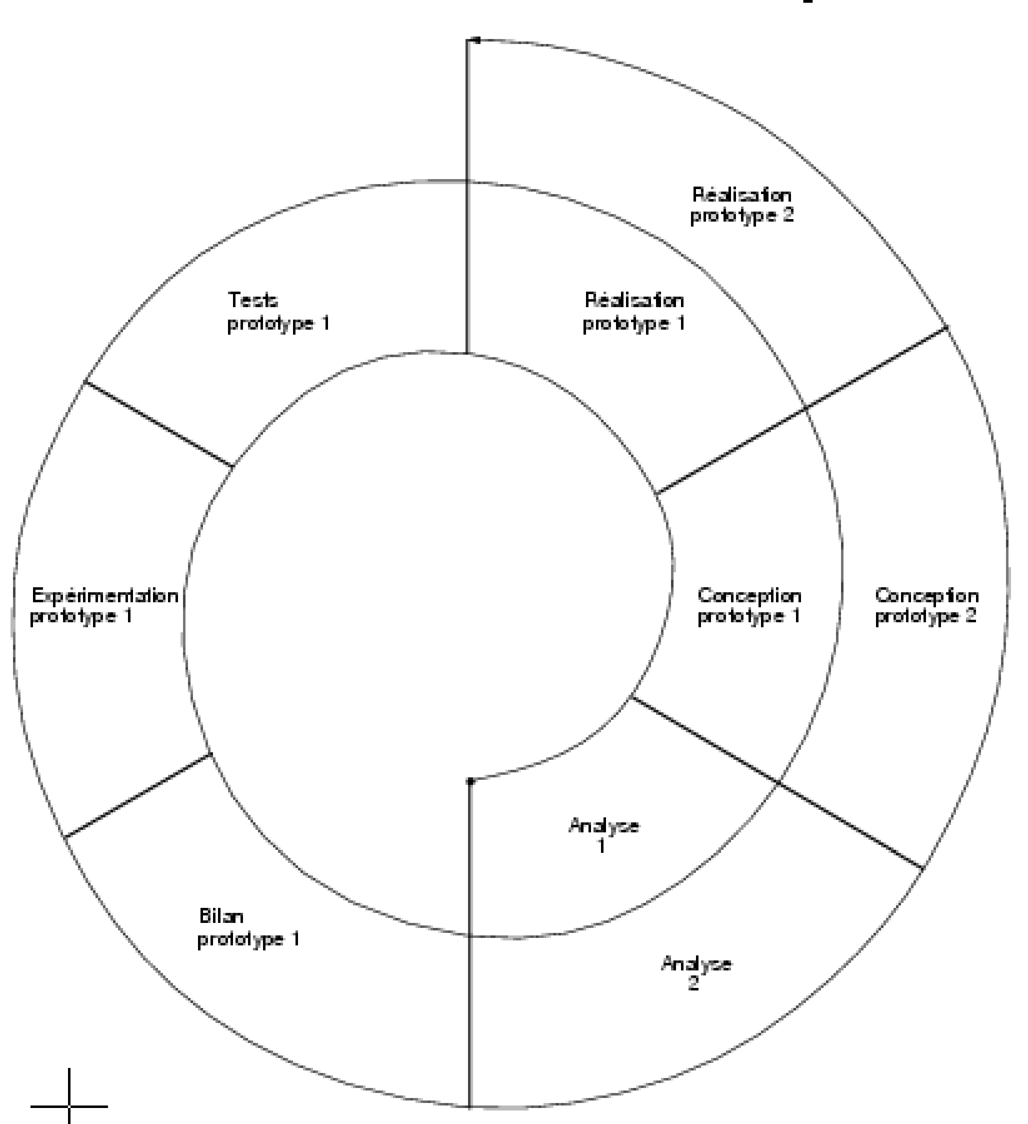
- <u>Gestion de projet</u> : pilotage du projet
- <u>Gestion des configurations</u> : gestion des différentes versions du produit
- <u>Assurance qualité</u> : contrôle systématiquement que le produit en cours est cohérent et complet, en le confrontant à des normes préétablies si elles existent

Cycles de développement en W



- Maquette : défilement d'écrans donnant une idée de ce que sera la future application (sans accès aux données)
- Les maquettes sont élaborées par les informaticiens et validées par les utilisateurs
- Avantages du maquettage
  - Gain de temps sur les phases en aval (2nd V)
  - Limitation des erreurs lors de la recette

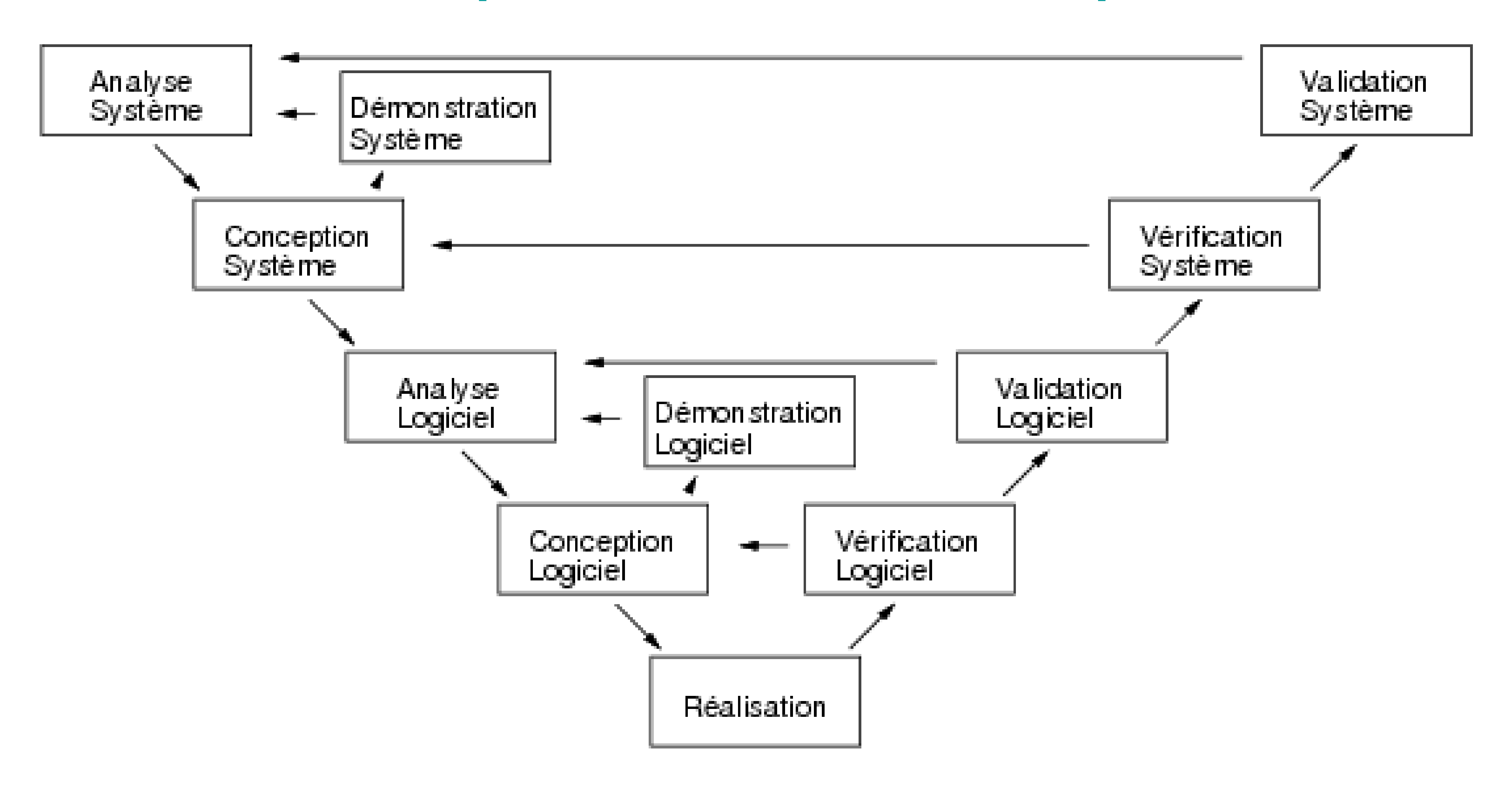
Cycles de développement en spirale



- <u>Prototype</u> : application en réduction (avec accès aux données)
- <u>Expérimentation</u> : tests de la part des utilisateurs du produit dans sa version actuelle (éventuellement définitive)
- <u>Bilan</u> : critique de l'expérimentation

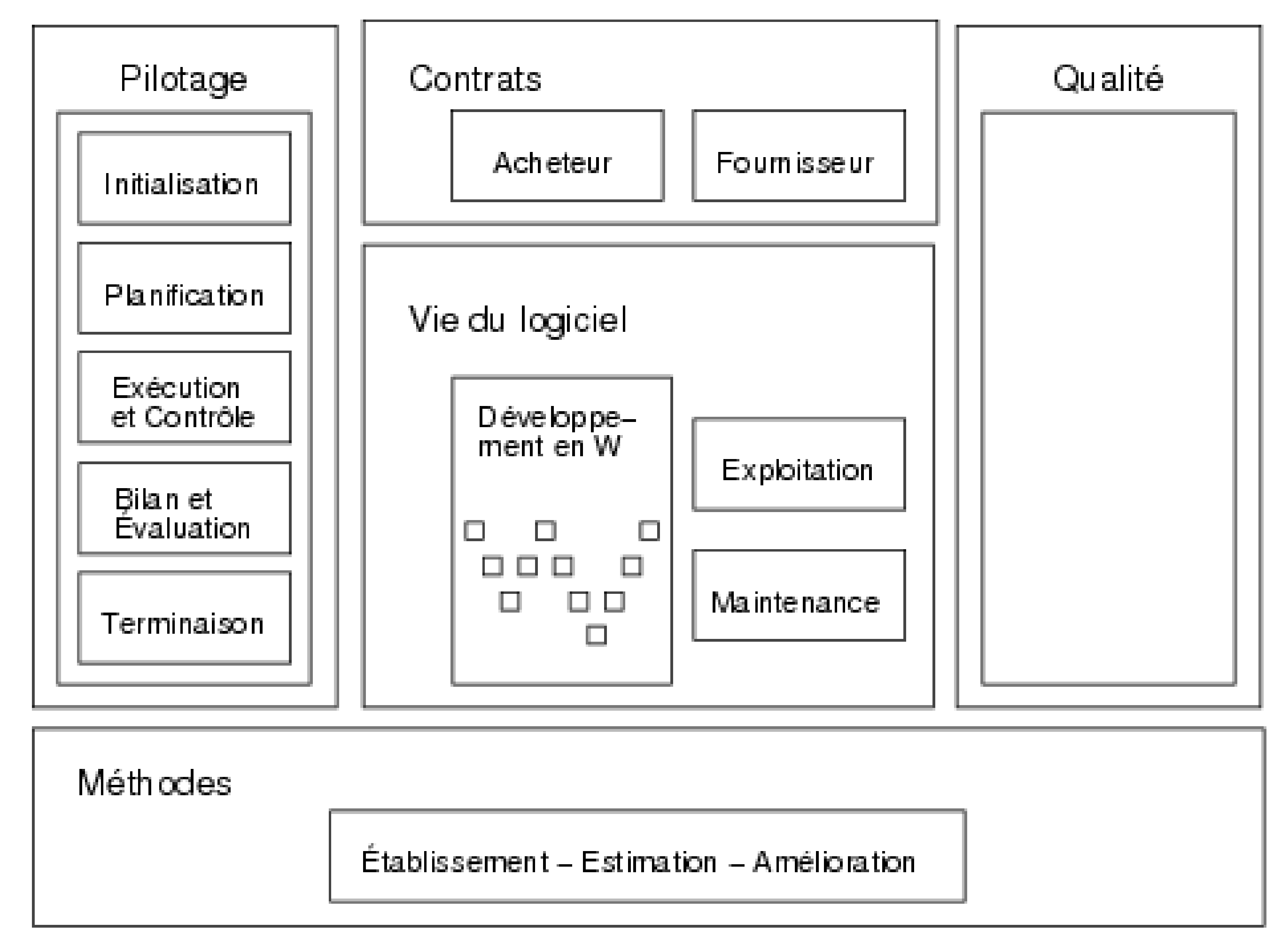
- Généralisation de l'approche par itération
- Ex. : conception d'outils de pilotage (car une forte réactivité aux besoins non stables des utilisateurs est nécessaire)

Cycles de développement composite : un exemple

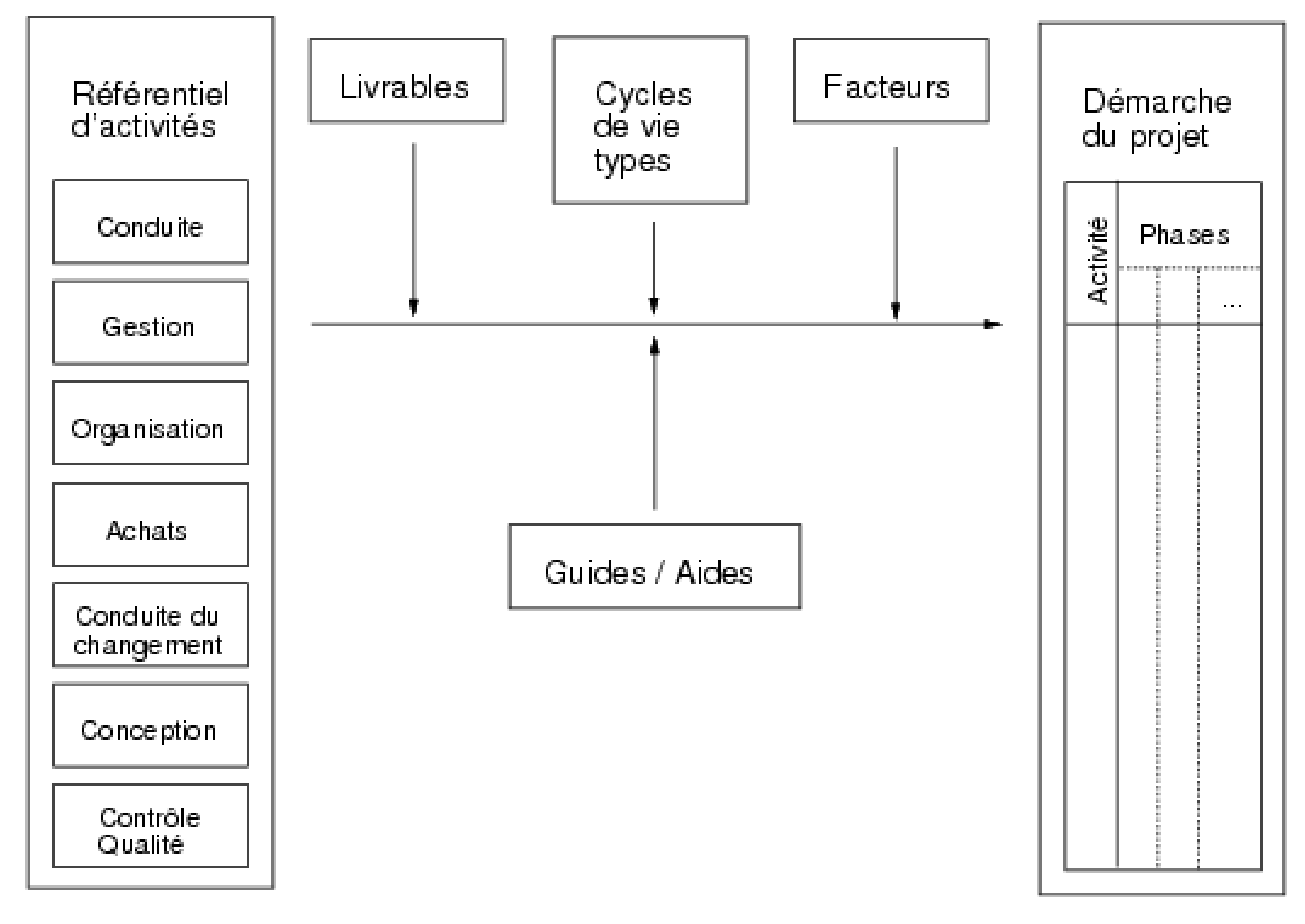


<u>Démonstration</u> : présentation du produit aux utilisateurs

Cycles de vie de l'ISO



Cycles de vie d'EuroMethode



Chiffres : coût moyen relatif de chaque phase (du cycle de développement du logiciel) pour une application de gestion

• Analyse et Conception :	44 %
• Réalisation :	28 %
• Tests :	28 %

Chiffres : coût relatif de correction d'une erreur selon la phase (du cycle de vie du logiciel) au cours de laquelle elle a été détectée

Analyse :	1
Conception :	2
Réalisation :	5
Tests :	10
Exploitation et Maintenance :	plus de 100

- Remarque : plus de 80 % des erreurs sont introduites durant les phases d'analyse et de conception

- Les coûts de la maintenance corrective (ni adaptative, ni évolutive) peuvent aller jusqu'à deux fois ceux du développement
  Exemple pathologique (système avionique) : coût de développement de 30$ par instruction mais coût de maintenance de 4000$ par instruction

Chiffres divers

- Productivité moyenne d'un programmeur d'une application de gestion simple : moins de 600 lignes de code par mois

- Application moyenne (en 1985) : 100 000 lignes de code pour 600 000 €

    Ex. : suivi de production pour 3000 personnes, entreprise commerciale de 2 milliards de chiffre d'affaires

- Taille d’un projet
  - Entre 100 et quelques milliers de jours
  - Jusqu’à 50 personnes



Taxinomie des méthodes d’informatisation

Méthode d'informatisation : définition

Une méthode d'informatisation en informatique de gestion

– définit un processus d'informatisation du système d'information (totalement ou partiellement i.e. pour tout ou partie du cycle de vie du logiciel)
– possède une portée (champ d'étude i.e. domaine étudié)
– décrit une démarche i.e. un ensemble de travaux en les ordonnant (succession d’étapes)

Méthode d'informatisation : règles

- S'appuyer sur des concepts théoriques : définition des concepts
- Proposer une démarche : cadre général pour définir le travail à accomplir par les intervenants
- Permettre sa mise en œuvre par des outils : pour faciliter la manipulation des concepts
- Atteindre un but : l'informatisation éventuelle (⇒ argumentation et faisabilité)

N. B. : une méthode ne remplace ni l'expérience, ni la connaissance, ni le talent

Méthode d'informatisation : composants

- Modèles : ensemble de concepts et de règles destiné à expliquer et construire la représentation de phénomènes organisationnels
- Langages : destinés à l’élaboration des spécifications, à faciliter la communication
- Démarche
- Outils et techniques : aides à la mise en œuvre des modèles, langages, démarche

Méthode d'informatisation : objectifs

- Réduire la complexité des informatisations (ex. : en identifiant et donc en maîtrisant les facteurs de cette complexité)
- Rendre cohérents tous les projets (ex. : même approche, même « style » des dossiers, meilleure intégration entre projets)
- Capitaliser les expériences (ex. : réutilisation des solutions ayant résolu les mêmes problèmes, acquisition de savoir-faire)
- Augmenter la qualité des travaux d'informatisation (ex. : mêmes standards)
- Augmenter la productivité des travaux d'informatisation (ex. : standardisation augmente l'efficacité)
- Améliorer les communications entre intervenants (utilisateurs et informaticiens)

Méthode d'informatisation

N. B. : les SSII ont été les premières à créer des méthodes

Les solutions empiriques

- Avantage : répondent à l'urgence
- Inconvénient : génèrent des applications provisoires (car complexes, non fiables, coûteuses, etc.)

Taxinomie des méthodes : fondements théoriques

- Cartésienne (démarche dite analytique ; résolution des problèmes un à un)
  - Approche fonctionnelle (analyse et conception des systèmes d'information par rapport à la définition des besoins) et descendante (du général au particulier)
  - Ex. : SADT, CORIG

- Systémique (démarche dite globalisante ; résolution globale des problèmes)
  - Approche conceptuelle (processus de modélisation par niveaux d'abstraction successifs)
  - Repose sur l'identification de projets qui structurent l'organisation (sans qu'il y ait obligatoirement un besoin)
  - Ex. : MERISE, AXIAL, IA-NIAM

Taxinomie des méthodes : fondements théoriques

- À objet (application du paradigme objet à tout le processus)
  - Les objets (de l'application, de services distribués) et les utilitaires communs échangent des informations (demandes et réponses de services) à l'aide de messages
  - Ex. : OOA, OMT, MCO, HOOD, OOSE, MERISE Objet
- Formelle (utilisation des mathématiques)
  - Spécification et conception formelles exprimées à l'aide du langage mathématique qu'il faut ensuite prouver
  - Ex. : B

Taxinomie des méthodes : générations

- Première génération
  - Des années 60 au début des années 70
  - Automatisation des procédures administratives
  - Problèmes de programmation (ex. : WARNIER, JACKSON, etc. sur l'art de bien écrire du code i.e. programmation structurée)
  - Approche analytique (par les données) ou synthétique (par les fonctions)
  - Ex. : MINOS (analytique), CORIG (synthétique)

Taxinomie des méthodes : générations

- Deuxième génération
  - Années 70
  - Généralisation des champs d'étude au système d'information et à l'organisation en entier
  - Préconisations (en analyse, conception, programmation) et démarche d'informatisation (schéma directeur, plan d'informatisation, conduite de projet)
  - Prise en compte de nouvelles techniques (temps réel, bases de données, ergonomie), nouvelles formalisations (entités-associations), évolution des sciences de gestion
  - Ex. : IA-NIAM, SADT

Taxinomie des méthodes : générations

- Troisième génération
  - Depuis la fin des années 70 (dont les méthodes à objets des années 80)
  - Informatisation globale (cohérence, complétude)
  - Innovations technologiques (matérielles et logicielles)
  - Démarche de synthèse, davantage de modélisation, introduction d'outils logiciels associés
  - Ex. : MERISE, AXIAL, SSADM, OOA, OMT, OOSE, HOOD, B

- Quatrième génération ?
  - Intégration des technologies orientées objets, client/serveur

Taxinomie des méthodes : domaines d'application

- Particulier
  - Application à un travail précis et indépendant de toute démarche
  - Ex. : RACINES pour l'élaboration d'un schéma directeur
- Partiel
  - Description et ordonnancement de travaux relativement à une démarche d'informatisation partielle
  - Ex. : CORIG pour la conception et la réalisation du système informatique, SADT et IA-NIAM pour la conception du système d'information et du système informatique
- Global
  - Processus d'informatisation complet (de l'introduction de l'informatique dans une organisation à la maintenance des applications) : description et ordonnancement de tous les travaux
  - Ex. : MERISE, AXIAL, SSADM, OOA, HOOD, OMT, OOSE, B

Taxinomie des méthodes : démarche

- Linéaire
  - Succession linéaire des travaux (démarche découpée en étapes découpées en phases découpées en tâches découpées en opérations)
  - Analyse descendante (des problèmes généraux aux problèmes particuliers) par décomposition hiérarchique des travaux
  - Itération et condition possibles
  - Ex. : MERISE, AXIAL, etc.
- Non linéaire
  - Analyse ascendante par intégration progressive des résultats

Taxinomie des méthodes : approche

- Ascendante
  - Recensement et analyse des sorties (papier ou écran) puis établissement des entrées nécessaires et suffisantes
  - La liste des informations obtenue est ainsi l'ensemble minimal nécessaire pour obtenir les résultats, ce qui permet difficilement de prendre en compte l'évolution des besoins de l'organisation
  - Ex. : MINOS
- Descendante
  - Recensement des informations du système d'information existant (sans oubli ni répétition) et des nouvelles fonctionnalités des utilisateurs
  - Ex. : CORIG, MERISE, etc. (la plupart des méthodes actuelles)

Quelques méthodes : ≤1982

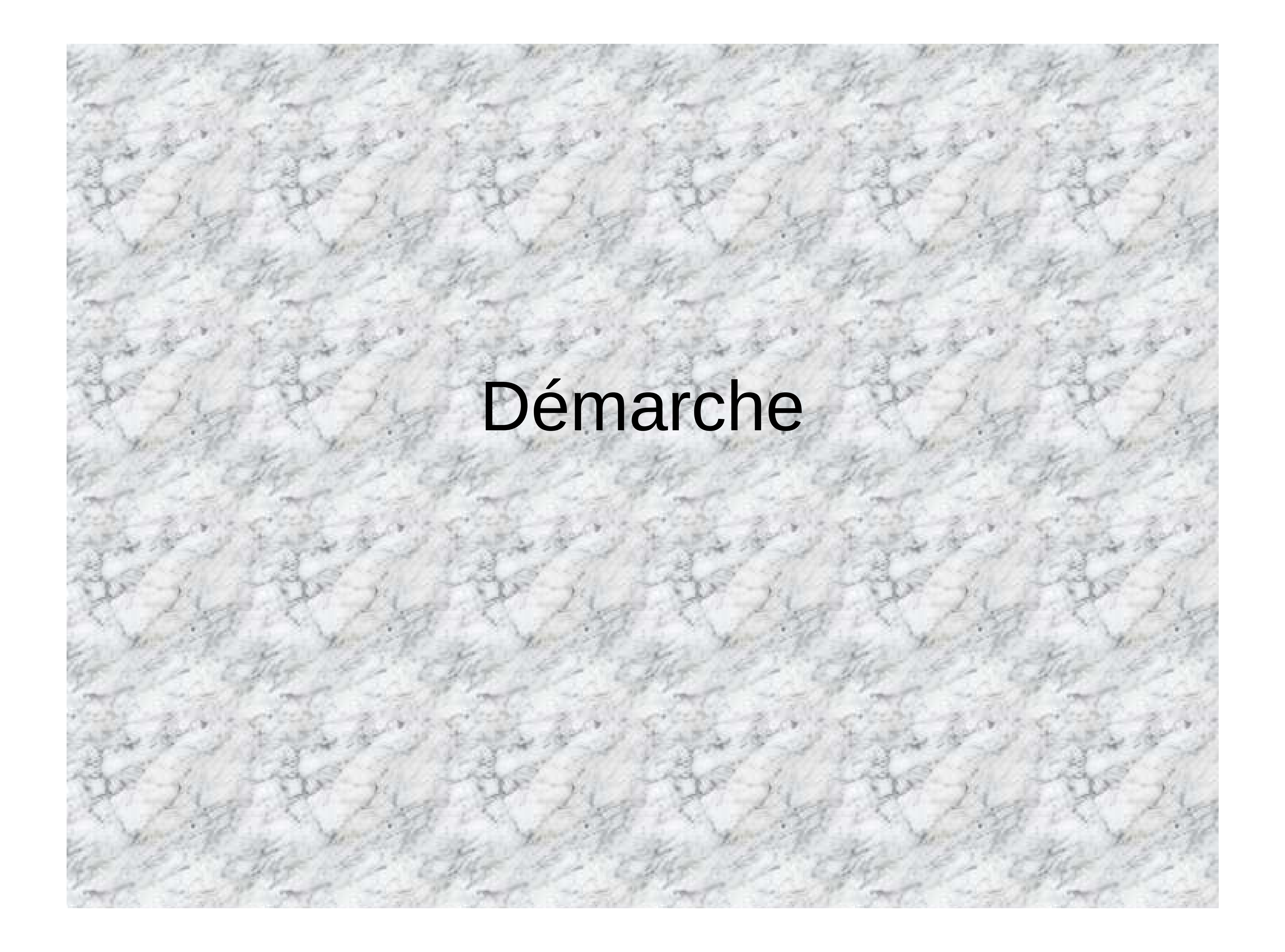
- CORIG
  - Compagnie Générale d'Informatique, France, 1966
  - Conception et réalisation du système informatique
- SADT (Structured Analysis and Design Techniques)
  - D.T. ROSS pour SofTech, USA, 1976 (et IGL Technology, France, 1977)
  - Conception du système d'information et du système informatique
- MCP (Méthode de Conduite de Projets informatiques)
  - RATP et AFCET, France, 1978
  - Conduite de projets
- [H]OOD ([Hierarchical] Object Oriented Design)
  - R. ABBOTT en 1980, G. BOOCH en 1983 (CISI & MATRA & CRI pour l'Agence Spatiale Européenne en 1987)
  - Conception et réalisation du système informatique
- IA-NIAM (Nijssen's Information Analysis Method)
  - M. NIJSSEN pour Control Data, Belgique, 1982
  - Conception du système d'information et du système informatique

Quelques méthodes : 1983..1986

- MCX et MCO (Méthode générale d'analyse des applications informatiques)
  - X. CASTELLANI, France, 1983
  - Informatisation complète
- MERISE et MERISE/2 (Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise)
  - H. TARDIEU pour Séma-Matra et Gamma International, France, 1983
  - Informatisation complète
- JSD (Jackson System Development)
  - M. JACKSON, Systems Ltd, Royaume-Uni, 1983
  - Conception du système d'information et du système informatique
- SSADM (Structured Systems Analysis and Design Method)
  - LBMS pour CCTA, Grande-Bretagne, 1986
  - Informatisation complète
- AXIAL (Analyse et Conception de Systèmes d'Information Assistés par Logiciels)
  - IBM, France, 1986
  - Informatisation complète

Quelques méthodes : ≥1988

- REMORA
  - C. ROLLAND de l'Université Paris 1 (Sorbonne), France, 1988
  - Conception du système d'information et du système informatique
- OOA (Object-Oriented Analysis)
  - P. COAD et E. YOURDON, 1991
- OMT (Object Modeling Technique)
  - J. RUMBAUGH, 1991
  - Conception du système d'information et du système informatique
- OOSE (Object-Oriented Software Engineering)
  - I. JACOBSON, 1992
- Z
  - D. LIGHTFOOT, 1992
  - Conception du système d'information et du système informatique
  - N. B. : c'est un langage de notation et non pas une méthode
- B
  - H. HABRIAS, 1993
- UML (Unified Modeling Language)
  - G. BOOCH, I. JACOBSON et J. RUMBAUGH, 1999
  - N. B. : c'est un langage de modélisation et non pas une méthode


Démarche

Démarche d’une méthode d'informatisation traditionnelle

- Étude préalable
- Analyse fonctionnelle
- Analyse organique
- Programmation
- Mise en service

Démarche : 3 premières étapes

- Étude préalable
  - Étude de l’existant → *dossier de l’existant validé*
  - Étude d'opportunité → *rapport d'opportunité*

  → *cahier des charges* (et *plan directeur de réalisation*)
- Analyse fonctionnelle
  - Conception (modèles de communication, des traitements et des données)
  - Validation

  → *schéma conceptuel*
- Analyse organique
  - Progiciel ou Développement spécifique

  → *solution informatique*

Démarche : étude préalable (objectif)

- Analyse du fonctionnement de l'organisation et diagnostic général de l’existant
- Recensement des critiques (positives ou négatives, d’organisation et informatiques) et des besoins des utilisateurs
- Opportunité (financement, moyens humains, etc.) et faisabilité (technique) des automatisations
- Rédaction d'un cahier des charges

Démarche : étude de l’existant (importance)

Toute l'application en dépend $\Rightarrow$

- exhaustivité
- exactitude

Gravité croissante d'une étude préalable se révélant incomplète ou inexacte lors de l'analyse fonctionnelle et/ou organique (peu grave), de la programmation (dommage), de l’exploitation (catastrophique)

Démarche : étude de l’existant (objectif)

Description de l'existant (par différentes représentations littéraires/schématiques et modèles de communication/traitement/données) en collectant toutes les informations (informatisées ou non) utiles et nécessaires

Démarche : étude de l’existant (phases)

- Collecte
  - Aller sur le terrain
  - Observer
  - Questionner
  - Prendre des notes
  - etc.
- Représentation
  - Rédiger
  - Formaliser les renseignements collectés
  - Modéliser
  - etc.
- Validation

Démarche : collecte

- Objectif : recueillir et sélectionner les informations intéressantes (i.e. pertinentes) parmi toutes les informations vues (i.e. observées) ou entendues (via entretiens)
- Informations à recueillir
  - Nature, volume, fréquence, précision observée ou requise, durée de vie, ancienneté, etc.
  - Exemplaires vierges et renseignés

Démarche : collecte (critères)

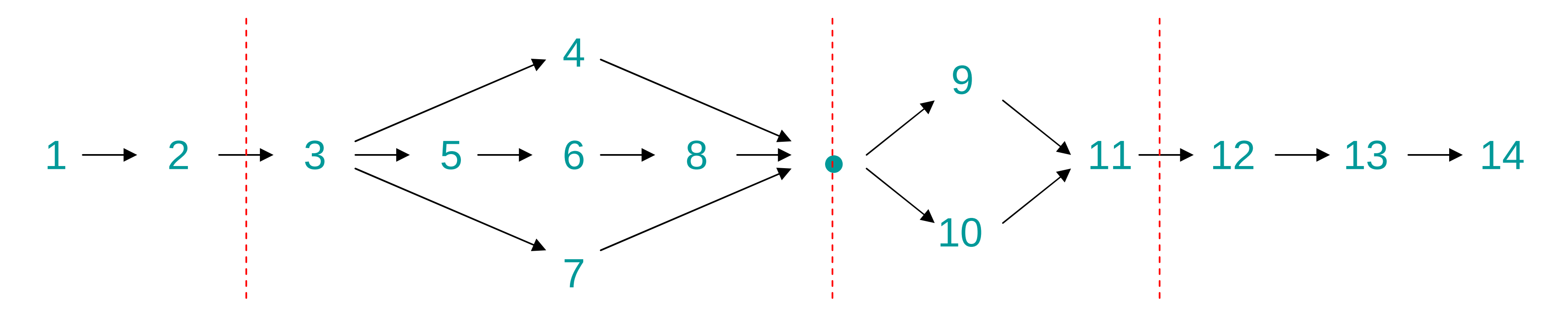
- Informations sur le système actuel et futur
- Informations sur le système ou du système
  - Ne recueillir que les informations directement utiles liées à l'étude
- Informations de type
  - Dynamique : circulation des documents dans l’espace (ex. : diagramme de circulation des documents ou de l’information, diagramme de flots de données) et dans le temps (calendrier, temps des traitements, délai de circulation, etc.)
  - De transformation : procédure de traitement, règle de gestion, enchaînement des tâches, formule de calcul, condition de déclenchement des traitements
  - Statique : données élémentaires (ex. : dictionnaire des données) et documents (fiches de rubriques/fichiers/documents), services et postes de travail (ex. : organigramme, fiche de fonction)
- Degré de conscience ou d’expression de l'information
  - Collecter les informations exprimées (par écrit ou oralement)
  - Détecter les informations conscientes non exprimées
  - Deviner les informations inconscientes
  - N. B. : selon le cas, faire exprimer/reconnaître les informations non exprimées ou les laisser dans l'ombre

Démarche : collecte (moyens)

- À partir de documents (écrits et collectés)
  - Documents existants : d’exécution (ex. : facture, bulletin de paye, bordeaux, fichiers produits, etc.), de gestion (ex. : organigrammes, statistiques, etc.) ou à établir entièrement
  - Documents à compléter (questionnaire)
- Entretien (ou enquête orale)
  - Accompagnant des documents écrits (pour les expliquer/compléter/contrôler/mettre à jour) ou sans document écrit préalable (avec ou sans la participation de l’interlocuteur)
  - Contraignant ou peu directif (selon expérience/aisance de l’analyste)
  - Quelques conseils : fixer un rendez-vous, préparer l’entretien, être ponctuel, préciser l'objectif, questionner, écouter, noter, demander tous les documents nécessaires, conclure, faire un compte-rendu
- Observation (ou enquête visuelle)
  - Après un entretien par exemple
  - Qualitative (sur le déroulement d’une procédure d’un poste de travail, sur la circulation empruntée par un document marqué, etc.) ou quantitative (ex. : mesurer le nombre de tâches pour une période donnée, la durée d’exécution d’un travail, etc.)

Démarche : collecte (ordonnancement des tâches)

Tâches d'introduction (définition de l'étude)
Tâches d'analyse du présent (recueil de l'existant)
Tâches montrant les contraintes et désirs de l'organisation future (critique)
Tâches de conclusion



Remarques

- Il existe des tâches séparées dont la collecte est commune, et inversement une tâche peut nécessiter des collectes séparées
- Avancées ou retour en arrière possibles
- Il ne s'agit que d'un ordonnancement possible

Démarche : collecte (ordonnancement des tâches d'introduction)

1. Prise de connaissance du contexte i.e. de la structure hiérarchique de l'organisation et de son environnement social, technique et économique
2. Reformulation des limites de l'étude et du découpage en projets à partir de ce qui a été décrit ou demandé

Démarche : collecte (ordonnancement des tâches d'analyse du présent)

3. Au niveau du projet retenu, étude de la structure hiérarchique et liste des postes de travail et des centres de décision
4. Étude détaillée des postes de travail
5. Établissement d'une liste des fichiers et des documents
6. Représentation de la circulation des documents mentionnant les traitements
7. Recensement et description des règles de gestion (:= condition facultative, affectation et règle de calcul), i.e. les procédures et règles de traitement
8. Confection d'un dictionnaire des rubriques

Démarche : collecte (ordonnancement des tâches montrant les contraintes et désirs de l'organisation future)

9. Récapitulation des moyens et ressources utilisés et des contraintes (durée, délai, fréquence, volume, coût, réglementation, ergonomie)
10. Récapitulation des demandes d'information et des critiques formulées par le personnel consulté
11. Contrôle du travail effectué i.e. des éléments du système d'information existant répertoriés au cours de l'analyse

Démarche : collecte (ordonnancement des tâches de conclusion)

12. Constitution du dossier de l’existant i.e. première version du cahier des charges détaillé
13. Validation de l'étude auprès des personnes compétentes et concernées par l'étude
14. Premier examen critique des personnes ayant réalisé cette analyse mentionnant leurs avis sur l’existant

Démarche : étude d’opportunité (objectif)

Faciliter la prise de décision par la direction générale en commission informatique sur la suite à donner à l'étude (par un rapport synthétique présentant les principales critiques formulées et les diverses solutions envisageables) i.e. la mise en œuvre d'un certain nombre de projets d'automatisation parmi ceux proposés

Démarche : étude d'opportunité (critique du système d'information existant)

- Niveaux : général, des domaines d'étude, des services et postes de travail
- Causes possibles de dysfonctionnement
  - Insuffisance des moyens de traitement de l'information (ex. : en personnel, matériel, locaux)
  - Mauvaise organisation (ex. : centralisation excessive ou insuffisante, personnel inadapté ou incompétent, mauvaise structure hiérarchique)
  - Circuits informationnels mal étudiés (ex. : trop longs, non compris)
  - Méthodes de traitements mal formalisées ou archaïques (ex. : inexistence d'algorithme)
  - Documents inexistants ou inutiles ou incomplets
  - Fichiers inexistants, mal structurés, incomplets, redondants, etc.
- Exposé des besoins nouveaux exprimés par les utilisateurs

Démarche : étude d'opportunité (propositions de solutions)

- Pallier les dysfonctionnements et améliorer le système
  - Solutions non informatisées
    - Personnel (ex. : embauche, promotion, déplacement, formation)
    - Matériels (ex. : achat, remplacement, entretien, déplacement)
    - Documents (ex. : création, modification, suppression, amélioration du circuit)
    - Méthodes (ex. : réorganisation des tâches, définition des algorithmes)
    - Fichiers (ex. : création, restructuration)
  - Solutions informatisées
    - Définition des tâches devant être automatisées
    - Découpage en projets d'automatisation homogènes et relativement indépendants, en faisant apparaître les priorités de réalisation

Démarche : étude d’opportunité (synthèse des propositions de solutions)

- Évaluation financière (coût estimé et gain escompté) de chaque proposition
- Présentation de l'ordre des priorités entre les différentes solutions
- Mesure de la faisabilité et établissement de la mise en œuvre (en prenant en compte des mesures d’accompagnement : personnel, matériel, logiciel, etc.) de chaque proposition

Plan directeur de réalisation

Présentation de toutes les modalités de réalisation (des programmes d'application spécifiques) :

- responsabilités
- personnel d’exécution
- calendrier de réalisation de chaque projet
- liaisons entre les différents projets (ou logiciels acquis)
- interventions éventuelles d'informaticiens extérieurs à l'organisation

Cahier des charges

- Destinataires

  Service informatique, constructeur ou société de services en informatique

- Objectif

  Définir les besoins en matériel et en logiciel du futur système informatique (pour permettre de choisir l'une des solutions) afin d'établir un contrat entre utilisateurs et informaticiens

Cahier des charges
(renseignements informatiques)

- Description détaillée des fonctionnalités attendues
- Évaluation chiffrée des volumes à mettre en œuvre
  - Données à stocker, sauvegarder, saisir, imprimer
  - Modes de travaux envisagés : immédiat ou en temps différé, unitaire ou par lot
    Ex. : saisie d'un questionnaire, édition des commandes du jour, sauvegarde incrémentale, édition préprogrammée des bulletins de paye
  - Nombre maximum d'utilisateurs connectés simultanément

Cahier des charges
(renseignements informatiques)

- Définition des besoins en matériel
  - Types de postes de travail (bureau, ordinateur, taille écran, type d'imprimante, etc.)
  - Réseau de communication utilisé (privé ou public, en étoile ou en bus ou en anneau ou …, local ou global, etc.)
  - Périphériques particuliers (ex. : lecteur de code à barres)
- Définition des besoins en logiciel
  - Progiciels systèmes (système d'exploitation, compilateurs et interpréteurs des langages de programmation, utilitaires, gestionnaire des données, gestionnaire réseau, etc.)
  - Progiciels d'application

Cahier des charges (renseignements technico-commerciaux, avant la livraison)

- Conditions financières des matériels et logiciels : achat, location, maintenance, etc.
- Conditions d’extension de la configuration matérielle (mémoire principale, mémoire auxiliaire, périphériques, etc.) et des logiciels (amélioration des performances, volumes de données, nouvelles fonctionnalités, etc.), en assurant portabilité et compatibilité
- Conditions d'implantation des matériels (plan, onduleur, climatisation, puissance électrique, etc.) et logiciels (ex. : système d’exploitation, mémoire minimale, etc.)
- Conditions d’essais (des performances, sur du matériel équivalent, etc.)
- Conditions de livraison (délai, pénalités de retard, responsable du transport du matériel, installation et adaptation, support logiciel, livraison partielle, etc.)

Cahier des charges (renseignements technico-commerciaux, après la livraison)

- Conditions de maintenance (durée, jour et délai d’intervention, coût des nouvelles versions, etc.)
- Durée d'utilisation en cas de location
- Formation du personnel (nature, durée, date, coût, lieu, etc. des stages et cours)
- Aide à la mise en œuvre (durée et périodicité, nombre de techniciens à disposition, etc.)
- Documentation (nature, coût, nombre d’exemplaires, nouvelles versions, etc.)
- Conditions de reconversion des applications existantes

Démarche : analyse fonctionnelle

- Objectif

  Obtenir un schéma général de structuration des traitements et des données, à un niveau conceptuel (c'est-à-dire indépendant de tout matériel ou logiciel de base)

  Quoi faire ?

- Critères d'un schéma conceptuel
  - Communicable (avec utilisateurs et autres informaticiens)
  - Conforme
  - Valide : complet et cohérent
  - Réalisable (automatisable en partie)

- Principe d'indépendance des traitements et des données

  Pour cela, un logiciel (le SGBD) doit être capable, au moment de l'exécution des programmes, de retrouver les données nécessaires aux traitements à effectuer

  Indépendance logique (respectivement physique) lorsque le schéma conceptuel (respectivement logique) des données peut être modifié sans changer les programmes

Démarche : conception

- Représentation de la communication au sein de l'organisation
- Représentation de l'ensemble des traitements
  - Modélisation des traitements avec leurs conditions d'activation, leurs règles d'utilisation et de transformation, leur enchaînement, etc.
    - Statique : description d'un traitement
    - Dynamique : spécification des conditions d'exécution (événement déclencheur) et d'enchaînement (en séquence, en parallèle, convergent, etc.) de traitements pour caractériser le comportement du système
- Représentation de l'ensemble des données
  - Modélisation de toutes les informations (et de leurs structures) devant être manipulées (et donc stockées)
  - Contraintes d'intégrité : conditions à satisfaire pour les données mémorisées par le système d'information
    - Statique : vérifiées à tout moment
    - Dynamique : caractérise la validité des changements d'états du système d'information

    Remarque : certaines contraintes sont déjà incluses dans les modèles

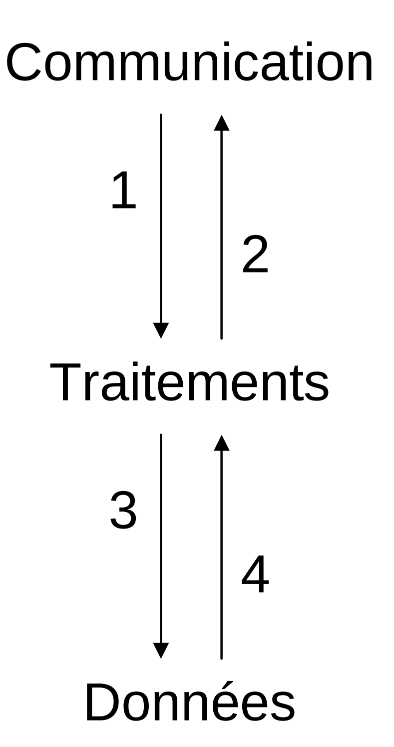
Démarche : validation

- Validation formelle des traitements et des données
  - Complétude des traitements
    - L’ensemble des traitements décrits correspondent à la définition
  - Cohérence des traitements
    - Statique : pas de contradiction
    - Dynamique : pas d'inter-blocage, terminaison
  - Complétude des données
    - Pas d'oubli (respect de la norme décrivant le modèle)
  - Cohérence des données
    - Conformité à la norme : pas d'ambiguïté, pas de contradiction, pas de redondance, désagrégation/décomposition

Démarche : validation

- Synthèse des différents schémas (de communication, des traitements, des données) garantissant la cohérence du schéma conceptuel

1. Toute communication s'appuie (si besoin) sur un traitement
2. Tous les traitements assurent les communications de l'organisation avec son environnement et en son sein
3. Aucun traitement ne fait référence à une donnée n'existant pas
4. Toutes les données sont manipulées par au moins un traitement



Démarche : validation

- Confrontation avec les utilisateurs

Démarche : analyse organique (objectif)

Adapter la solution fonctionnelle à un choix technique particulier

- Définition des structures de données et de leur enregistrement
- Détermination des unités de traitement
- Choix des matériels
- Établissement du calendrier et des budgets de réalisation

Démarche : progiciel vs développement spécifique

- Achat d'un progiciel standard (i.e. un PGI)
  - Plus économique (à long terme)
  - Présent sur de nombreux segments de marché
  - Produit déjà testé
  - S'assurer de la réelle adaptation aux besoins
  - Complexité du paramétrage
  - Peut nécessiter de recourir à un spécialiste
  - Existe-t-il un service après-vente, de plus viable à long terme ?
- Développement spécifique
  - Solution parfaitement adaptée aux besoins
  - Deux approches : traditionnelle ou génie logiciel
  - Deux étapes
    - Étape logique : choix d'organisation (*Qui fera quoi ? Où ? Quand ?*)
    - Étape physique : choix techniques (*Comment : avec quels moyens matériels et logiciels ?*)

Démarche : approche traditionnelle

- Représentation des traitements
  - Étape logique : prise en compte des contraintes des utilisateurs faisant intervenir le temps (date de début au plus tôt, date de fin au plus tard, durée, date de début effective, etc.) et le lieu (communication entre les acteurs, poste de travail effectuant le traitement, traitement manuel ou interactif ou différé) des traitements

    Cas particulier : procédures de fonctionnement en mode dégradé (données détruites, lieu ou ressource indisponible)
  - Étape physique : fait intervenir les contraintes de ressources nécessaires et utilisées (regroupement de traitements successifs, éclatement d'un traitement) et affecter les responsabilités des traitements

Démarche : approche traditionnelle

- Représentation des données
  - Étape logique : prise en compte des besoins d'utilisation des informations (ex. : définition des modes d'accès aux données)
  - Étape physique : prise en compte des contraintes physiques liées en particulier aux matériels et logiciels utilisés (ex. : description des données par rapport à leur implantation, calculs d'activité afin de déterminer les schémas/vues/index/clusters/redondances/etc. les plus efficaces, etc.)

Démarche : approche traditionnelle

- Structure d'accueil : mémoire, processeur, réseau, langage, progiciel, etc.
- Interface homme-machine : ergonomie, langage de communication
- Méthode de conception : analyse descendante
- Programmation : programmation objet (encapsulation, héritage, polymorphisme, etc.)
- Calendrier
- Budget de programmation

Démarche : approche génie logiciel (objectif)

Passer à l'ère industrielle de la production du logiciel, en développant des méthodes et des techniques permettant de réaliser à moindre coût des logiciels performants et fiables

Démarche : approche génie logiciel

- Concevoir (le produit)
  - Résultat d'une analyse ou d'une étude de marché
  - Fournir un ensemble de spécifications détaillées
  - Choisir une interface utilisateur : graphique (on évitera dorénavant une interface en mode texte)

Démarche : approche génie logiciel

- Fabriquer
  - Principe : décomposer en composants plus simples, et mettre au point un processus d'assemblage
  - Pour chaque composant identifié, trois choix possibles :
    - Utiliser un composant standard : SGF ou SGBD, bibliothèque mathématique, bibliothèque de classes, applets JAVA, contrôles VBX ou ActiveX, etc.
    - Le fabriquer soi-même
    - En sous-traiter la fabrication, lorsque les coûts sont trop importants, par une entreprise spécialisée
  - Implémenter les traitements : décomposition modulaire
  - Implémenter les données : données transitoires et permanentes
  - Méthode : programmation descendante, objet

Démarche : approche génie logiciel

- Langage de programmation
  - Choix d'un paradigme (ex. : procédural, déclaratif, fonctionnel, L4G, objet, etc.)
  - Identification des besoins : objets, systèmes répartis, bases de données, systèmes concurrents, etc.
  - Identification des moyens : disponibilité du produit sur les plateformes cibles, personnel formé
- Choix des outils
  - Outils de développement rapide : pour du prototypage car les performances sont souvent insuffisantes
  - Générateurs de code : description de haut niveau des traitements à réaliser, code généré en L3G
  - Outils spécialisés : SGBD, gestionnaire réseau, architecture client/serveur, etc.

Démarche : approche génie logiciel

- Tester
  - Jeux d'essais : jeux de données couvrant tous les cas possibles, générateurs de tests
  - Simulation du fonctionnement : injection de données, *brenchmark* de systèmes
  - Tests en grandeur nature (par les utilisateurs finals)

Démarche : approche génie logiciel

- Prouver/Valider
  - Méthodes mathématiques de preuve de programmes (cf. FLOYD, HOARE, etc.)
  - Preuve des spécifications formelles du logiciel
  - Utilisation d'outils de validation
  - Vérifier l'adéquation aux besoins

Démarche : approche génie logiciel

- Évaluer les performances
  - Calculs des complexités *a priori* (et s'assurer que les charges des machines suffiront)
  - Tests en grandeur nature (dans l’environnement final, dans les conditions réelles d’exploitation)

Démarche : approche génie logiciel

- Assurer la fiabilité
  - Plus aucune erreur majeure
  - Risque d'erreurs (mineures) résiduelles (ex. : conditions limites non testées)
  - Révisions successives du logiciel (versions alpha, béta, release, mises à jour mineures et majeures)

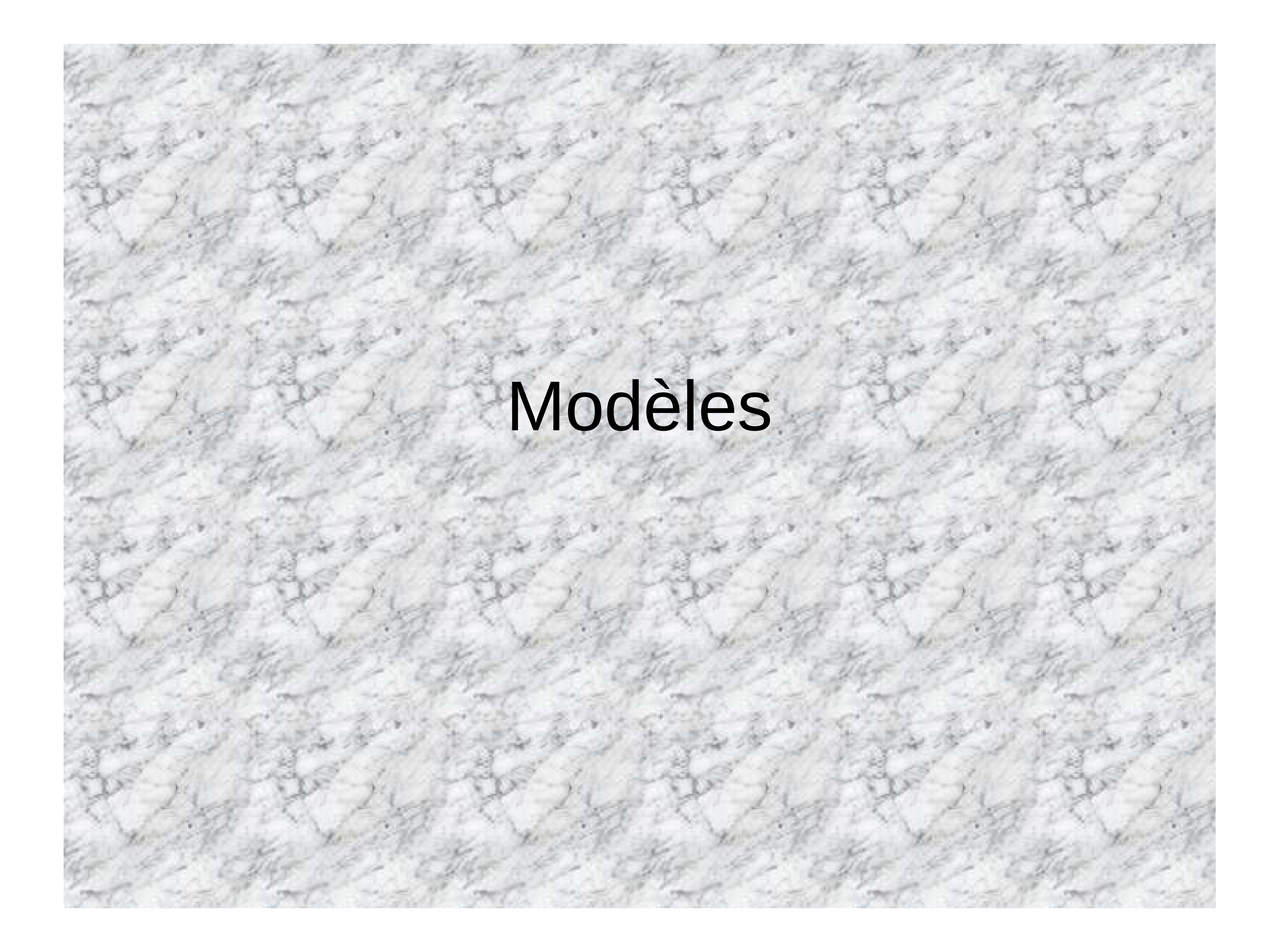
Démarche : approche génie logiciel

- Fournir une documentation
  - Technique
    - Durant tout le développement (*dossier de programmation*) : communication entre sous-équipes, rédigé quotidiennement par les développeurs
    - Pour la maintenance (*guide de maintenance*) : recherche ultérieure des causes d'erreurs
    - Pour l'installation (*guide d'installation*)
  - Utilisateur
    - Mode d'emploi pour un produit sur mesure (*manuel utilisateur*) : précis, technique, sans fioritures
    - Communication pour un produit grand public : rédigé par des professionnels

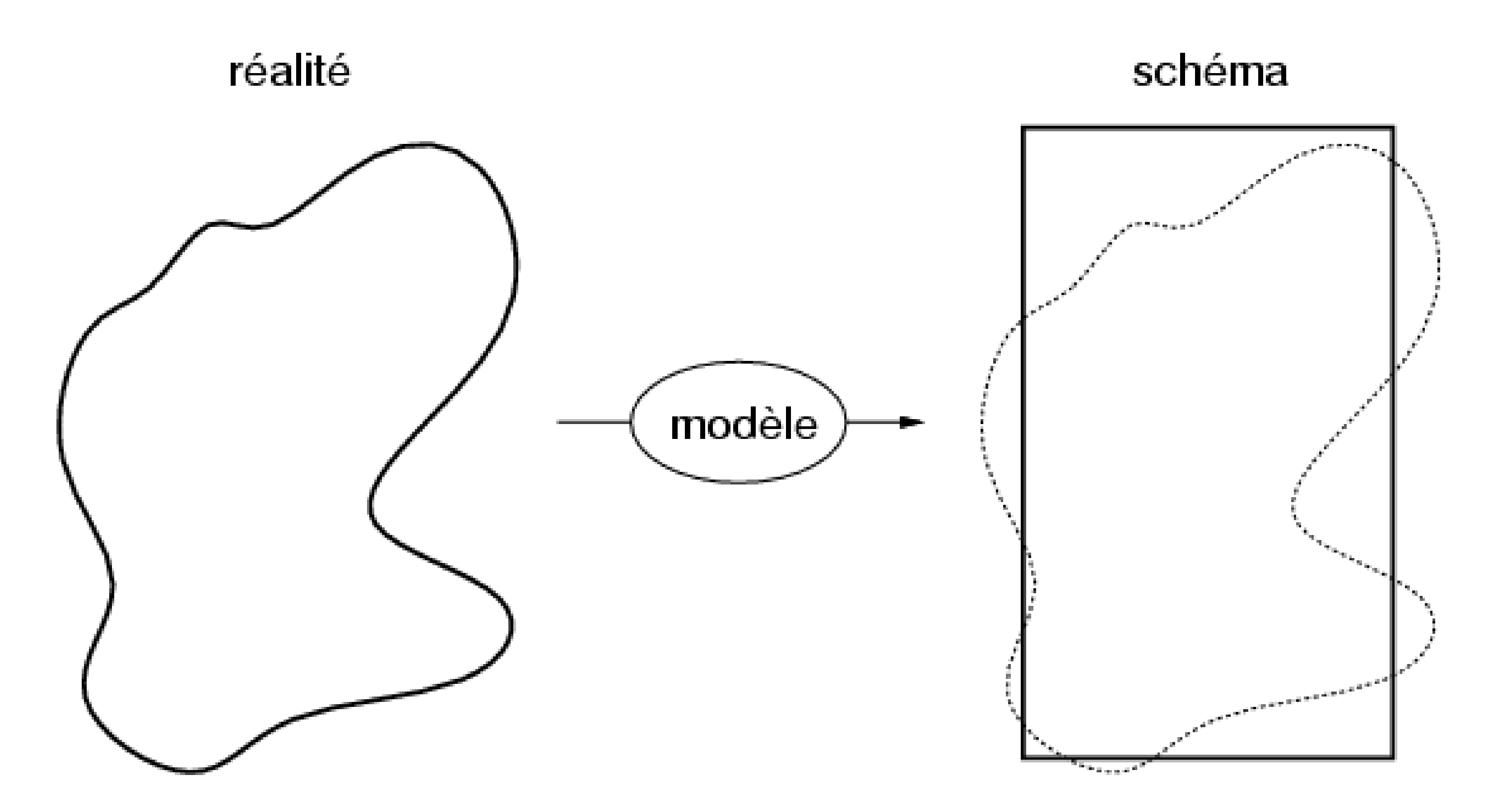
N. B. : précise, claire, fiable, à jour, etc.

Démarche : approche génie logiciel

- Proposer un service après-vente
  - Maintenance sur site, ligne directe, service payant
  - Formation


Modèles

Modèles : modélisation



- Problème de la réalité : flou, difficile à appréhender, etc.
- Deux types d'erreurs : réalité omise et schéma prenant en compte davantage que la réalité
- Rarement un seul modèle (union de modèles)

Modèles : objectifs de la modélisation

- Rendre compte de la réalité
  - Conforme
  - Complet
  - Réalisable
  - Plausible
- Simplifier la réalité
- Ne présenter qu'un aspect du problème
- Permettre de mieux comprendre un problème complexe
- Permettre de communiquer les connaissances
  - Standard
  - Précis
  - Simple
  - Cohérent

Modèles : outils et types

- Outils
  - Langage naturel
  - Représentation graphique
  - Mathématiques
- Types de modèles
  - De communication, des traitements ou de données
  - Statique ou dynamique

Modèles : l’exemple « jouet »

- Les traitements
  - Le jour de la rentrée, le secrétariat de l’établissement avise les étudiants qu'ils ont jusqu’à la fin de la semaine pour amener les originaux des diplômes qu'ils ont obtenus, ceci permettant de compléter les fiches des étudiants
  - Un mois après la rentrée, le secrétariat transmet une photocopie des fiches au directeur de l’établissement
- Un fichier (… de cinq étudiants)

Modèles : l'exemple « jouet »

Numéro INE de l'étudiant : 2
Nom : LEROI
Département de naissance : 40

Voitures possédées par l'étudiant :

Numéro d'immatriculation	*Couleur*

Diplômes obtenus par l'étudiant :

Intitulé abrégé	*Intitulé complet*	*Année*
BAC	Baccalauréat	1980
DEUG	Diplôme d'Études Universitaires Générales	1982

Numéro INE de l'étudiant : 3
Nom : DUPOND
Département de naissance : 17

Voitures possédées par l'étudiant :

Numéro d'immatriculation	*Couleur*

Diplômes obtenus par l'étudiant :

Intitulé abrégé	*Intitulé complet*	*Année*
BAC	Baccalauréat	1981
DUT	Diplôme Universitaire de Technologie	1983
MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises	1985

Modèles : l'exemple « jouet »

Numéro INE de l'étudiant : 4
Nom : MARTIN
Département de naissance : 47

Voitures possédées par l'étudiant :

Numéro d'immatriculation	*Couleur*
4747 LA 47	rouge

Diplômes obtenus par l'étudiant :

Intitulé abrégé	*Intitulé complet*	*Année*
BAC	Baccalauréat	1977
DEUG	Diplôme d'Études Universitaires Générales	1980
MIAGe	Maîtrise des Méthodes Informatiques Appliquées à la Gestion des Entreprises	1982

Numéro INE de l'étudiant : 5
Nom : DURAND
Département de naissance : 33

Voitures possédées par l'étudiant :

Numéro d'immatriculation	*Couleur*
3333 BX 33	rouge
4040 NT 40	jaune

Diplômes obtenus par l'étudiant :

Intitulé abrégé	*Intitulé complet*	*Année*
BAC	Baccalauréat	1981
DUT	Diplôme Universitaire de Technologie	1983

Modèles : l’exemple « jouet »

Numéro INE de l’étudiant : 7
Nom : LEROI
Département de naissance : 33

Voitures possédées par l’étudiant :

Numéro d’immatriculation	*Couleur*

Diplômes obtenus par l’étudiant :

Intitulé abrégé	*Intitulé complet*	*Année*

- Problème des zones variables (zéro, une ou plusieurs valeurs) engendrant des difficultés de stockage efficace
  - Ex. : les voitures des étudiants
- Problème de redondance
  - Ex. : intitulé complet des diplômes

Quelques modèles

- Actigramme
- Algèbre relationnelle
- Arbre de décomposition fonctionnel
- Calcul relationnel
- Cycle de vie d'un objet
- Datagramme
- Diagramme d’activités
- Diagramme d’états-transitions
- Diagramme d’objets
- Diagramme de cas d’utilisation
- Diagramme de circulation de l'information
- Diagramme de circulation des documents
- Diagramme de classes
- Diagramme de collaboration
- Diagramme de communication
- Diagramme de composants
- Diagramme de déploiement
- Diagramme de flots de données

Quelques modèles

- Diagramme de flots d'événements entre classes
- Diagramme de flots d'événements entre objets
- Diagramme de séquence
- Diagramme de structure
- Diagramme de structures composites
- Diagramme de suivi d'événements
- Diagramme de temps
- Diagramme des paquetages
- Diagramme d'états (structuré)
- Diagramme global d'interaction
- Fiche de description de fonction
- Fiches de description de document
- Fiches de description de fichier
- Fiches de description de rubrique
- Graphe acteurs-flux
- Grille d'analyse des rubriques
- Logique des propositions et des prédicats
- Machine abstraite

Quelques modèles

- Modèle conceptuel des traitements
- Modèle conceptuel des traitements analytique
- Modèle de flux (modèle de contexte, modèle de flux conceptuel, modèle de flux organisationnel)
- Modèle dynamique
- Modèle entités-associations (ou modèle conceptuel des données)
- Modèle fonctionnel
- Modèle logique des données
- Modèle logique des données réparties
- Modèle logique des traitements (guidage fonctionnel, interface utilisateur (présentation, dialogue), noyau applicatif non interactif)
- Modèle logique des traitements répartis
- Modèle navigationnel
- Modèle objet
- Modèle organisationnel des données
- Modèle organisationnel des traitements
- Modèle organisationnel des traitements analytique
- Modèle relationnel

Quelques modèles

- Organigramme
- Réseaux de Pétri
- Schéma d'architecture logique des moyens informatiques
- SQL
- Substitution généralisée
- Table de décision
- Théorie des ensembles et relations



Méthodes et langages de modélisation

MERISE/2
SADT
OMT
UML
B